



International Journal of Computer Science & Emerging Technologies

ISSN 2044-6004 (Online)

Volume 1 Issue 2

August, 2010

© Sprinter Global Publication, 2010

www.sprinterglobal.org/ijcset

ijcset@gmail.com

IJCSET BOARD MEMBERS

Editor in Chief

I.Khan, UK

N. Aslam, UK

Editorial Members

- **Sattar B. Sadkhan**, USA
- **Khoa N. Le**, Australia
- **Sasan Adibi**, Canada
- **Mostafa M. El-Said**, USA
- **Nidaa A. Abbas**, IRAQ
- **Eleonora Pantano**, Italy
- **Yin-Fu Huang**, Taiwan
- **TAIB Nabil**, ALGERIA
- **Dilip S. Aldar**, India
- **John Woods**, UK
- **Ing. Reuben Farrugia**, Malta
- **Pankaj Kumar**, India
- **T V Narayana Rao**, India
- **N. RADHIKA**, India
- **N. Sridhya**, India
- **Osman Ismail**, Saudi Arabia
- **Sultan Aljahdali**, Saudi Arabia
- **Mohammed Badawy**, Egypt
- **Munir Ahmed**, Saudi Arabia
- **M Abdul Rahman**, India
- **Raiz Ul Islam**, South Korea
- **Nafiz Imtiaz Bin Hamid**, Bangladesh
- **Jasvir Singh**, India
- **Manas Ranjan Biswal**, India
- **Ratnadeep R. Deshmukh**, India
- **Sujni Paul**, India
- **Mousa Demba**, Saudi Arabia
- **Yasser Alginahi**, Saudi Arabia
- **Tarun Kumar**, India
- **Dr. Alessandro Agostini**, Saudi Arabia
- **Ka Lok Man**, China
- **Priti Srinivas Sajja**, India
- **Samy El-Tawab**, USA
- **Baoning WU**, USA
- **Constantin Volosencu**, Romania
- **Shanyu Tang**, UK
- **Georgios Dafoulas**, UK
- **Jun Peng**, USA
- **Ion Mierlus Mazilu**, Romania

- **Smita Rajpal**, India
- **Messaouda Azzouzi**, Algeria
- **M.Ramakrishnan**, India
- **A.Arul Lawrence selvakumar**, India
- **Arun Sharma**, India
- **A.V.Senthil Kumar**, India
- **Mieczyslaw Drabowski**, Poland
- **Bilal Alatas**, Turkey
- **Prasant Kumar Pattnaik**, India
- **Subramanian Karthikeyan**, Sultanate of Oman
- **Siti Zaiton Mohd Hashim**, Malaysia
- **Mohammad Nazir Ahmad**, Malaysia
- **Shang GAO**, Canada
- **Azween Bin Abdullah**, Malaysia
- **Alicia Nicki**, USA
- **Kenneth Revett**, United Kingdom
- **Florin Gorunescu**, Romania
- **Marina Gorunescu**, Romania
- **Riktेश Srivastava**, UAE
- **Padmaraj Nair**, USA
- **Abdel-Badeeh M.Salem**, Egypt
- **Donald Hsu**, USA
- **P Kiran Sree**, India
- **M.Ramakrishnan**, India
- **Prabhat K. Mahanti**, Canada
- **Karthikeyan Subramanian**, Sultanate of Oman
- **Mieczyslaw Drabowski**, Poland
- **Shang Gao**, Canada
- **Gongjun Yan**, USA
- **Priti Srinivas Sajja**, India
- **Siti Zaiton Mohd Hashim**, Malaysia

TABLE OF CONTENTS

1. A Scheme for a Secure Decision Support System of Semiotics Type (pp-1:7)

Ayman M. Brisha

Faculty of Computer science and engineering, Taibah University, Saudi Arabia

2. Study of Advances in Real-Time Simulation of Hybrid Electric Vehicles (pp-8:13)

Sachin Kumar Suryvanshee

Department of Electrical Engineering, Jabalpur Engineering College Jabalpur India

Mr. Arun Pachori

Department of Electrical Engineering, Jabalpur Engineering College Jabalpur India

3. Performance of Reversible Image Watermarking Algorithm using Histogram Shifting under Noisy Conditions (pp-14:24)

S. Kurshid Jinna

Department of Computer Science & Engineering PET Engineering College Vallioor, India

Dr. L. Ganesan

Department of Computer Science & Engineering A.C College of Engineering & Technology, India

4. A Novel Image Compression Algorithm based on Discrete Wavelet Transform with Block Truncation Coding (pp-25:34)

Shyam Lal

Department of E & C Engineering, Moradabad Institute of Technology, India

Mahesh Chandra

Department of E & C Engineering, Birla institute of Technology, Mesra-Ranchi (Jharkhand), India

Gopal Krishna Upadhyay

SSITM, Kasganj, Moradabad (U.P.), India

5. Expert System For Online Diagnosis of Red-Eye Diseases (pp-35:39)

Muhammad Zubair Asghar

Institute of Computing and Information Technology, Gomal University, Pakistan

Muhammad Junaid Asghar

Department of Basic Medical Sciences, Gomal University, Pakistan

6. Handwritten Devnagari Numeral Recognition using SVM & ANN (pp-40:46)

Sandhya Arora

Department of CSE & IT, Meghnad Saha Institute of Technology, Kolkata, India

Debotosh Bhattacharjee, Mita Nasipuri, M. Kundu, D. K. Basu

Department of Computer Science and Engineering, Jadavpur University Kolkata, India

L.Malik

Department of Computer science, G.H. Rasoni college of Engineering, Nagpur, India

7. Real Time Wireless Sensor Network for Coastal Erosion using Fuzzy Inference System (pp-47:51)

Arabinda Nanda

Department of Computer Science, Krupajal Engineering College, Bhubaneswar, India

Amiya Kumar Rath

Department of Computer Science & IT, College of Engineering, Bhubaneswar, India

Saroj Kumar Rout

8. SVM Classifier Technique for Fingerprint Based Gender Identification (pp-52:57)

Arun K.S

Department of Computer Science and Engineering, St. Joseph's College of Engineering, Kerala, India

Sarath K.S

Department of Electronics and Communication, College of Engineering Kidangoor, Kerala, India

9. Performance Analysis Of Lossless Compression Schemes For Bayer Pattern Color Image Video Sequences (pp-58:62)

G. Mohanbabu

Department of Electronics and Communication Engineering, Dindigul, India

Dr. P. Renug

Department of Electrical and Electronics Engineering, Madurai, India

10. A Framework for Semantic Web Services Discovery using Improved SAWSDL-MX (pp-63:71)

Agushaka J. O

Department of Mathematics, Ahmadu Bello University Zaria-Nigeria

Junaidu S. B

Department of Mathematics, Ahmadu Bello University Zaria-Nigeria

11. A Novel Approach towards Cost Effective Region-Based Group Key Agreement Protocol for Ad Hoc Networks using Chinese Remainder Theorem (pp-72:82)

K. Kumar, J.Nafeesa Begum

CSE, Government College of Engg, Bargur, Tamil Nadu, India

Dr.V. Sumathy

ECE, Government College of Technology, Coimbatore, Tamil Nadu, India

12. A Migrating Parallel Exponential Crawling Approach to Search Engine (pp-83:90)

Jitendra Kumar Seth

Department of Information Technology, Ajay Kumar Garg Engg. College, Ghaziabad, India,

Ashutosh Dixit

Computer Science Department, YMCA, Faridabad, Hariyana, India

13. Web Page Prediction Model Based on Clustering Technique (pp-91:94)

Rajni Pamnani

Department of computer technology, VJTI University, Mumbai, India

Pramila Chawan

Department of computer technology, VJTI University, Mumbai, India

14. A Novel Approach to Face Detection using Blob Analysis Technique (pp-95:99)

D.P.Tripathi

Roland Institute of Technology, BPUT, Orissa, India

S.N.Panigrahy

Gandhi Institute of Industrial Technology, BPUT, Orissa, India

Dr.N.P.Rath

Veer Surendra Sai University of Technology, Orissa, India

15. Survey on Multimedia Operating Systems (pp-100:108)

P. DharanyaDevi, S. Poonguzhali, T. Sathiya, G.Yamini, P. Sujatha and V. Narasimhulu

Department of Computer Science, Pondicherry Central University, Pondicherry, India

16. Survey on Distributed Operating System: A Real Time Approach (pp-109:123)

Shailesh Khapre, Rayer Jean, J. Amudhavel, D. Chandramohan, P. Sujatha and V. Narasimhulu

Department of Computer Science, Pondicherry Central University, Pondicherry, India

17. Mapping and Generation Model for Named Entity Transliteration in CLIR Systems (pp-124:133)

V. Narasimhulu, P. Sujatha and P. Dhavachelvan

Department of Computer Science, Pondicherry Central University, Pondicherry, India

18. Study and Improvement on Optimal Reactive Routing Protocol (ORRP) for Mobile Ad-Hoc Networks (pp-134:139)

Soma Saha

Women's Polytechnic, Dept. of IT, Hapania, Tripura, India

Tamojay Deb

Department of IT, Dasaratha Deb Memorial College, Khowai, Tripura, India

19. Simulation Environments for Wireless Sensors Networks (pp-140:143)

Basavaraj.S.M

Appa Institute of Engineering and Technology, Gulbarga, Karnataka, India

V.D.Mytri

School of Computer Science and Engineering, XYZ University

Siddarama.R.Patil

P.D.A College of Engineering Gulbarga ,Karnataka, India

20. A Video Sharing Platform for mobile devices using Data Grid Technology (pp-144:154)

Sandip Tukaram Shingade, Pramila M Chawan

Computer Engg Department, VJTI, Mumbai, India

21. Predictive Preemptive Ad Hoc on-Demand Multipath Distance Vector Routing Protocol (pp-155:160)

Sujata.Mallapur

Appa Institute of Engineering and Technology, Gulbarga, India

22. A Novel Design of Multi-Port Cartesian Router (pp-161:167)

R.Anitha

Department of Electronics and Communication Engineering, PSNACET, Dindigul, India

Dr.P.Renuga

Department of Electrical and Electronics Engineering, TCE, Madurai, India

23. Ad-hoc Networking Applications in Different Scenarios (pp-168:174)

Md. Taslim Arefin

Department of Electronics and Telecommunication Engineering, Daffodil International University, Dhaka, Bangladesh

24. Commenting the Virtual Memory Management Kernel Source Code 2.6.31 for Educational Purpose (pp-175:178)

Archana S. Sumant

Veermata Jijabai Technological Institute (V. J. T. I.), Matunga, Mumbai, India

Pramila M.Chawan

Computer Technology Department, Veermata Jijabai Technological Institute (V. J. T. I.), Matunga, Mumbai, India

25. Efficient Service Retrieval from Service Store using Map Reduce (pp-179:185)

K.V. Augustine, S.K.V Jayakumar

Department of Computer Science, Pondicherry University, Puducherry, India

26. Handwritten Character Recognition Using Bayesian Decision Theory (pp-186:192)

Vijiyakumar , Suresh Joseph

Department of computer science, Pondicherry University, Pondicherry, India

27. A Relationship Oriented Framework for Learning in A Relationship Oriented Framework for Learning in a Structured Domain (pp-193:198)

Madhusudan Paul, Thamizh Selvam. D, P. Syam Kumar and Dr. R. Subramanian

Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India

28. Is Service Discovery necessary and sufficient – A Survey? (pp-199:205)

K.V. Augustine, E.Rajkumar

Department of Computer Science, Pondicherry University, Puducherry, India

A Scheme for a Secure Decision Support System of Semiotics Type

Ayman M. Brisha

Taibah University. Faculty of Computer science and engineering), Information systems dep, KSA
abrisha@taibahu.edu.sa

Abstract Semiotics refers to systems which can discover knowledge intelligently and help in decision-making. Algebraic semiotics provides a rigorous notation and calculus for representation that is explicitly value sensitive, while compassion supports both better analysis and better ethics in design. Semiotic Systems enable researchers to design beneficial and powerful systems. The contribution in this paper to enhance the Intelligent Decision Support Systems (IDSS) to the Secured Intelligent Decision Support Systems (SIDSS) to enhance his work. In this paper, the focus is on designing the coding model which encodes the representative sample of the data. The proposed (SIDSS) design is a coding base model which takes a sample representing the raw database from which processing can produce a secured knowledge base that helps making definitive system decisions in a short time. The proposed methodology provides the designer and developer specific guidance on the intelligent tools most useful for a specific user with a particular decision problem.

Keywords: *Decision Support Systems (DSS), Knowledge Discovery, Data Mining, Analytic Hierarchy Process (AHP), Intelligent Decision Making*

1. Introduction

Knowledge-Driven Decision Support Systems (KDDSS) can suggest or recommend actions to managers. Each KDDSS is a person-computer system with specialized problem-solving expertise. The expertise consists of knowledge about a particular domain, understanding of problems within that domain, with necessary skills for solving some of these problems[24]. A related concept is Data Mining, which refers to a class of analytical applications that search for hidden patterns in a database. Data mining is the process of sifting through large amounts of data to produce data content relationships. Tools used for building KDDSS are sometimes called Intelligent Decision Support Methods (IDSM). Many researchers such as in [24], have designed systems for implementing semiotic applications . Knowledge Discovery in Database (KDD) [34]. Provides organizations with the necessary tools to sift through vast data stores to extract knowledge, which supports and improves organizational decision making . KDD is defined as a nontrivial process of discovering useful knowledge from data [34]. The KDD process consists of such steps as data pre-processing (data selection – data cleaning and transformation), data mining

(i.e. extracting patterns such as classification rules extracted from a decision tree, that can support decision making [34]. Incremental data mining maintains patterns over a dynamic data source by revising patterns learned from a previous run of data mining, instead of learning from scratch. The value of information to the decision maker is often measured indirectly by evaluating information systems against some surrogate criteria. For example, the value of a decision support system (DSS) in the decision making process, and improvements in the outcomes from the use of the DSS [5-6]. However, none of these approaches provide a good measure of the decision value of DSS [5]. Several incremental data mining algorithms were proposed for major data mining models (as classification, association rule mining and clustering. For classification, algorithms ID4 and ID5 [34], were developed to revise a decision tree induced from old data as new data were added in. To support effective decision making , the KDD process needs to be completed. The KDD process cannot be fully automated, except for the data mining step. The success of the computer as a universal information-processing machine lies essentially in the fact that there exists a universal language in which many different kinds of information can be encoded and that this language can be mechanized [17]. Computers can compute, using binary notation for representing numbers is certainly of great interest, however there is nevertheless another key issue for making them able to process higher-levels of information. The first step in processing high level information was to code alphabetical symbols, therefore moving from the realms of numbers to the realms of words. The first binary encoding of alphanumeric characters was indeed designed nearly a century ago, by G.Peano [17], who is also responsible for the first axiomatization of arithmetic.

In section 2 discusses semiotic approach with algebraic semiotic. Section 3 explains the Intelligent Decision Support Systems and how we can evaluate it.

2. Semiotic Approach and Algebraic Semiotic

Semiotics is an interesting and powerful tool for rephrasing information theory and computer science. Semiotics provides a means of analysing the language of different healing modalities and our cultural understandings within which healing modalities are embedded [13]. It can also be used to show how social and political life may be shaped and influenced by the language we use to describe information and security. in this study, the semiotics are used to, focus on

a particular aspect of language, such as the metaphor, across a range of texts but the analysis tends to be quite broad and general. A semiotic approach to information systems provides a tool to represent organizational knowledge and activity. The theory of signs originates in the work of Peirce [12] who shows that a sign must be capable of evoking responses from the interpreter or a person. Semiotics makes us recognize the importance of an agent as being responsible for the existence of a sign and its meaning. Organizations can be seen as complexes of signs communicated among people who, acting as semiotic agents, are responsible for assigning meanings [1,5,12,14– 15 , 31]. Researchers have offered the concept of designing perspective intellectual systems as systems of semiotic type. The direction of an artificial intelligence named applied semiotics [13] has arisen and actively developed in the last few years. It unites researches in the field of semiotics modeling, semiotics knowledge bases, logic-linguistic and cognitive models, etc., which are necessary for the creation of highly effective intellectual systems (IS) capable of training (adaptation) and functioning in open and dynamic problem solving areas. The typical representative of such systems are intelligent decision support systems (IDSS) [2,32]. IDSS and Real-Time Support Systems (RT-IDSS) are intended to help decision makers manage complex objects and processes of various natures. These processes depend on conditions of hard time restrictions and concentrate on integrated intellectual systems. The primary goals decided by the RT IDSS are outlined in [16]. Forecasting involves drawing up a forecasting model of a progressing situation for an estimation of efficiency of recommended decisions for a particular solution ; Interaction with Decision Making (DM) (expert) - formation of the base of expert knowledge and delivery of the information (advice) to the DM. In the IDSS and RT-IDSS the data was treated as a whole, where any discrepancy in data or inaccuracy could lead to wrong decisions that may affect the system. This led to the proposed secured IDSS architecture in this paper in which only a sample of the data is used. The accuracy of the decision depends on the accuracy of the sample. Algebraic semiotic systems are a central notion of algebraic semiotics; describing axiomatic theories for systems of signs, including hierarchical "constructors" for signs, and (socially determined) measures of their relative importance. An example is the space of potential displays for some application running on the setting of a given sign, can be at least as important for meaning as the sign itself. On the contrary, the sentence "Yes" can mean almost anything, given an appropriate context. In algebraic semiotics, certain aspects of context dependency can be handled by constructors that place signs within larger signs, so that the original signs become contextualized sub signs. However, human interpretation is still needed for signs to have any meaning in any human sense [25]. Moreover, human interpretation is needed in deploying the formalism of algebraic semiotics, since it is intended to be used flexibly in musical performance [16]. Algebraic semiotics also provides precise ways to compare the quality of representations, and

to combine representations, such that conceptual blending (in the sense of cognitive linguistics is a special case as in [3]. Case studies for this theory include web-based displays for mathematical proofs that integrate motivation, background and explanation with formal details and information visualization [10]. It is difficult to design systems that satisfy users; failure is common, and even successful designs often overrun time and cost. Algebraic semiotics provides a rigorous notation and calculus for representation that is explicitly value sensitive, while compassion supports both better analysis and better ethics in design [16]. Algebraic semiotics help in solving the lack in scientific theories and support the design of virtual worlds, which are increasingly important in scientific research. In the next section the suggested technique for IDSS and its evaluation will be explained in detail.

3. Evaluation of Intelligent decision support systems (IDSS)

IDSS adds artificial intelligence (A. I.) functions to traditional DSS with the aim of guiding users through some of the decision making phases and tasks or supplying new capabilities. This notion has been applied in various ways. For example, [13] provided two layers in their framework for IDSS; a pragmatic layer associated with the actual performance of the task, and the conceptual layer associated with the processes and structure of the task. The study in [14] can be combined with other concepts to develop the IDSS architecture shown in Figure 1. Figure 1 illustrates an IDSS consisting of a data base, a knowledge base, and model base, some or all of which will utilize AI methods. The data base contains the data directly relevant to the decision problem, including the values for the states of nature, courses of action, and measures of performance. The knowledge base holds problem knowledge, such as guidance for selecting decision alternatives or advice in interpreting possible outcomes. The model base is a repository for the formal models of the decision problem and the approaches (algorithms and methodologies) for developing outcomes from the formal models. Decision-makers utilize computer and information technology to process the inputs into problem-relevant outputs. Processing will therefore involve: (a) organizing problem inputs; (b) structuring the decision problem decision model; (c) using the decision model to simulate policies and events; (d) finding the best problem solution. The IDSS can use knowledge drawn from the knowledge base to assist users in performing these processing tasks. Processing will generate status reports, forecasts, recommendations, and explanations. The status reports will identify relevant states, courses of action, and measures of performance and show the current values for these problem elements. Forecasts will report the states and actions specified in the simulations and the resulting projected values for the measures of performance. The recommendations are used to suggest the values for the actions that best meet the measures of performance.

Explanations will justify the recommendations and offer advice on further decision making. Such advice may include suggestions on interpreting the output and guidance for examining additional problem scenarios. Input feedback from the processing provides additional data, knowledge, and models that may be useful for future decision making. This feedback is provided dynamically to update the model and inputs in real time without external intervention. Output feedback is used to extend or revise the original analyses and evaluations. The literature provides numerous examples to show that IDSS can improve the decision making process and outcomes [9,19] . To provide a recent illustration of the use of both metrics, [38] has evaluated consumer DSS with the user’s cognitive effort to make and express preference in the decision processes and decision accuracy outcomes. IDSS supports cognitive tasks by playing an active role in aiding task performance, processing data and information to produce knowledge, and learning from experience [13]. They also support better decisions in terms of the outcome of the decision itself. The author propose that the “decision value” of IDSS should be evaluated by the effect on both the process of, and outcome from, decision making. Decision making in organizations and decentralized enterprises of today is increasingly distributed. Accurate and readily-available information can be provided through networked resources from a variety of sources and delivered to the decision maker in any location, and to a distributed group for collaborative decision making. Artificial intelligence enhances the potentialities of decision support systems in real management situations [27]. Hence, disparate resources are combined together and extend the support capabilities. In addition to IDSS improved outcomes, the use of AI techniques affects the process of decision making by providing the potential for real-time response, automation, personalization, sophisticated reasoning patterns, and broader information sources on which to base the decision. Intelligent systems achieve things differently than systems that do not embed intelligence. It is therefore appropriate, to specifically identify system benefits originating in process, as well as outcome support. The decision value of an IDSS, can therefore be determined from a multi-criteria evaluation using the process of, and outcome from, decision making as a top-level criteria.

4. The Analytic Hierarchy Process [AHP]

The analytic hierarchy process (AHP) is a multi-criteria method that can incorporate both qualitative and quantitative criteria into a single metric [28, 30]. Multi criteria decision making implies that a decision maker needs to identify the best course of action while considering a conflicting set of criteria. Complexity in decision making situations involves quantitative and qualitative criteria, multiple scales, and multiple comparisons. The ability to assign a preference rank for general decision making situations is needed as well as the simplicity of methods [29]. The AHP is a plausible method that provides a logical and scientific basis for such multi-criteria decision- making [11]. AHP has been widely

applied to both individual and group decision making scenarios from the early 1980s [30, 33]. According to [29], the AHP was founded on three design principles:(1) the decomposition of the goal-value structure where a hierarchy of criteria, sub criteria, and alternatives is developed, with the number of levels determined by the problem characteristics, (2) comparative judgments of the criteria on single pairwise comparisons of such criteria with respect to an upper criteria, and (3) linear-based synthesis of priorities where alternatives are evaluated in pairs with respect to the criteria on the next level of the hierarchy, with each criteria being assigned with a priority expressed as a weight in the AHP matrix. An advantage of the AHP for our evaluation of IDSS is that the contribution of the AI methods used in the system to individual criteria can be determined. For example, we can get more system process benefits by applying AI methods, or an AI method contributes to a specific phase of decision making. Such information assists the system developer as well as the user to understand the precise contributions of the components of the IDSS to the overall decision value. Previous studies have implemented the AHP to compare DSS and to determine their effect on the process of, and outcome from, decision making [5-7,18, 20-21]. In this research the study uses an evaluation of IDSS founded in [38], with the contribution of adding external memory to enhance the results founded in Table 1. The system performance can achieve the decision objective, for example, if the decision is intended to deliver decreased operating costs, then the organizational performance criterion is measured in terms of the cost decrease associated with the decision. Another possible outcome criterion shown in Table1 illustrates the growth in decision maker maturity.

Table 1 Weights assigned for the criteria in the AHP model

Level	Criteria by level	Weights	Comments
Decision making	Process/outcomes to decision value	[0.40, 0.60]	User consider outcome more important than process (60% for 40%)
	(Decrease in redundant complaints/precision of decision making to outcome)	[0.70, 0.30]	User considers the decrease in redundant complaints more important than precision of decision making (70% vs. 30 %)
	(Intelligence /design/choice/ learning proficiency to process)	[0.20, 0.50, 0.20, 0.10]	The user considers the design of the infrastructure solution more important than other phases
Decisional service task	(Analysis /synthesis to intelligence)	(0.80, 0.20)	For the intelligence phase, the user considers the support provided by analysis to be most important.
	(Analysis /synthesis to design)	(0.10, 0.90)	For the design phase, the user considers synthesis to be most important
	(Analysis	(0.50, 0.50)	For the choice phase,

	/synthesis to choice)		the user considers both analysis and synthesis of equal importance
	(Analysis /synthesis to learning)	((0.30, 0.70)	For the learning phase, the user considers synthesis to be more important
	(Analysis /synthesis to decrease in redundant complaints)	(0.75, 0.25)	The user considers analysis to be important in the decrease in redundant complaints
	(Analysis /synthesis to precision of decision making)	(0.30, 0.70)	The user considers synthesis to be more important in the precision of decision making
Architectura l capability	(User interface/D&K/ processing to analysis services)	(0.10, 0.45, 0.45)	The user considers D&K and processing to be important for analysis
	(User interface/D&K/ processing to synthesis services)	(0.10, 0.20, 0.70)	The user considers processing to be most important for synthesis

Figure 2 illustrates an AHP model for IDSS evaluation. The Decision Value of the IDSS is at the top of the hierarchy and depends on the decision process and outcome. The outcome describes the achievement by the decision maker as a result of using the IDSS. Presumably, such learning would improve the decision making skills of the user in both the current and subsequent situations as found in[5]. The improvement can be measured by the user’s enhanced ability to perform decision making phases and steps, increased productivity (generating more alternatives and comparisons in a given time period), and enhanced efficiency (evaluating the same number of alternatives in a fixed time period). These improvements can be measured qualitatively (for example, self or expert ratings for decision task proficiency) and quantitatively (for example, productivity and efficiency in decision making). The process is described by the decision making phases of intelligence, design, choice, implementation and learning. As we move down the hierarchy, there is a Decisional Service-task Level, an Architectural-Capability Level, and finally a Computational-Symbolic Program Level with AI computational mechanisms as alternatives. The evaluator may choose to modify the AHP model to tailor the desired criteria for a specific IDSS[38]. The study in [38] has shown one possible implementation as shown in Figure 3, along with potential alternative AI methods including a genetic algorithms, intelligent agents, neural networks, a hybrid systems, or none meaning that no intelligence is embedded. In the AHP model, the alternatives are evaluated in pairs with respect to the three elements in the Architectural-Capability level: user interface, data & knowledge capabilities and processing capabilities. Thereafter, the alternatives are compared in this paper with respect to how well they provide personalization in the user interface. The user might indicate that agents are

much better than a neural network in providing personalization in the user interface, and this judgment is expressed in the AHP as a relative rating. An Eigen value computation is utilized to reconcile the pair wise judgments, and a ranking of alternatives on the specific criteria is produced using the judgments and the weighting of the criteria. The AHP then proceeds by evaluating the user interface, data and knowledge capabilities, and processing capabilities with respect to the three types of decisional services: analysis, synthesis or complex services. Numeric ranking values are produced for the alternatives using the criteria weights provided by the evaluator. The elements of the Decisional Service-task level are similarly evaluated with respect to the Decision Process level: intelligence, design, choice, implementation, and learning. Finally, a numeric ranking of the alternatives is computed for outcome, and these ratings are combined with the overall ratings calculated for process to provide an overall ranking of the alternatives with respect to the decision value representing, the highest level in the AHP model. The ranking at the top level indicates which alternative, has the best decision value, and a highest ranking can be interpreted as an selection of the best design for the IDSS. In addition, the precise contributions of each AI method for each criterion in the hierarchy can be determined.

5. Proposal for secured-IDSS design

The Architecture in figure 3 shows a generalized base architecture of an IDSS for the semiotics type. This architecture consists of the following nine blocks: problem analyzer, decision search, block of learning, raw database, model base, knowledge base, block of modeling, block of forecasting, and knowledge acquisition and accumulation. In the proposed method, the design of the Model-Base block is modified by using a Model coding instead of a Model Base. The newly proposed architecture is called a Secure IDSS (SIDSS). Throughout this study, this new architecture consists of the blocks shown in figure 3. The block of Modeling is classified by its output going to the block containing the table of coding where it is processed. Coding is being processed according to the type of data available and the output code is provided in the Model Coding. This modification contributes in securing information and keeping it confidential so no one can know the meaning of the data except those who have the right to examine the Table of Coding and understand the encryption code. The ability to understand the real meaning of the information helps in making decisions in addition to securing such data. SIDSS consists of two interfaces an environment and a user interface. The IDSS compiles and analyzes the data as well as creating models for the data. The IDSS also includes models to help in decision-making and other models used in training, modifying, and checking the data. Through models of knowledge, the IDSS can make accurate decisions. In this paper, the model base is replaced by a data encryption model that secures and protects information. The goal is to prevent information access / interpretation by unauthorized personnel

accessing the system. This approach helps in increasing overall accuracy and security.

6. The proposed scheme

This section considers features of the functions for a selection and updating in SIDSS. The SIDSS is a system of distributed intelligence organized by the principle of semiotics system, integrating in itself various adapted models of knowledge representation and search of the decision.

The SIDSS of semiotics type can be formally represented by the following set:

$$SS = \langle M, R(M), F(M), F(SS) \rangle$$

where $M = \{ M_1, \dots, M_n \}$ – is the set of the formal or logic-linguistic models which implement certain intellectual functions; $R(M)$ is the set of rules to choose the necessary model or set of models in the current situation. $F(M) = \{ F(M_1), \dots, F(M_n) \}$ is the set of rules for modifying the models $M_i, i=1, \dots, n$. $F(SS)$ is the rule of updating actually systems SS , namely its base designs $M, R(M), F(M)$ and, probably, itself $F(SS)$.

The monotony violation as a rules are use for Updating model conducts or switching from one model to another model. This switch is carried out by means of reaction to corresponding event or by means of performing certain rules, for example fuzzy conclusion rules such as; $A' \bullet (A \rightarrow B)$, where A' and A are the fuzzy sets describing conditions of problem area or object (the fuzzy relation of similarity between elements from A and A' should be determined), B is the fuzzy set of allowable models or modifications within the model. \bullet is a specified operation of a composition of fuzzy sets. One part to note is, that if corresponding sets of rules - for example, set of choice rules $R(M)$ - are production sets then they can be preliminary transformed into treelike structures such as decision trees or decision tables, that simplifies the choice procedure [32, 35-37].

7. Conclusion

The proposed scheme for Secured IDSS is a suitable system in applications for securing files, information and services on networks. The implementation of the design will be the next step in our research. SIDSS coding blocks combines both security and decision making which is the main advantage of this system, providing the security required in Decision Support Systems. The analysis of documents can be a very useful way of exploring some important social and political aspects of security. The analysis of documents can be a very useful way of exploring some important social and political aspects of security. There are obvious limitations in relation to the range of research questions you can ask

that documents will answer. The semiotic tool addresses the fundamental problem of reconciling differing perceptions within the organization to assist in overcoming the inherent problems of security. The suggested application case illustrated in this paper should be taken as a sample and initial effort to demonstrate the methodological design and evaluation potential capabilities of the proposed scheme.

References

1. J. Backhouse, "The use of semiotic analysis in the development of information system, PhD thesis, London school of Economics and Political Science (1992).
2. A.A. Bashlykov, Yeremeyev A.P. Ekspertnye sistemy podderzhki prinjatija reshenij v energetike (Expert decision support systems in the power engineering). - Moscow: Izd-vo MEI (MPEI Publishing), 1994. p. 216.
3. L. David Ritchie, "Metaphors of conceptual integration", *Metaphor and symbol* 19, 31-50, 2004.
4. G. Dhillon, interpreting the management of information systems security, PhD thesis, London school of Economics and Political Science (1996).
5. Forgionne, G., 1999. An AHP model of DSS effectiveness. *European Journal of Information Systems*, 95–106.
6. Forgionne, G., 2000. Decision-making support systems effectiveness: The process to outcome link. *Information Knowledge-Systems Management* 2, 169–188.
7. Forgionne, G., Kohli, R., 2001. A multiple criteria assessment of decision technology system journal qualities. *Information Management* 38, 421–435.
8. Gupta, N., Forgionne, G., Mora, M. (Eds.), 2006. *Intelligent Decision making Support Systems Foundations, Applications and Challenges*. Springer-Verlag, Germany.
9. Harker, P., 1988. *The Art and Science of Decision Making: The Analytic Hierarchy Process*. Working Paper 88-06-03, Decision Science Department, The Wharton School, University of Pennsylvania, Philadelphia, PA.
10. Joseph A. Goguen, "steps towards a design theory for virtual worlds", university of California, san Diego, 2004.
11. M. KarmÜftÜoğlu, Knowledge based information retrieval: A semiotic Approach PhD Thesis, City University, London (1998).
12. K. Kitiyadisai, concepts of Relevance in a semiotic framework Applied to information systems analysis and design, PhD thesis, London school of Economics and Political Science (1990).
13. H. Linger, F. Burstein, 1997. Intelligent decision support in the context of the modern organization. In: *Proceedings of the 4th conference of the international society for decision support systems – ISDSS'97*, Lausanne, Switzerland, July 21–22.
14. M. M. Marche, Models of Information: the feasibility of measuring the stability of data models, PhD Thesis, PhD thesis, London school of Economics and Political Science (1991).

15. S. Marche, on what a building might not be – a case study
International journal of information Management 11
(1991) 55-66.
16. Organizational Semiotics Workshop, 11-12 July 2003 at the
University of Reading, UK.
17. "The semiotics of the web", Philippe eodognet, universite
paris6, 4 palace jussive, 75005 paris France, 05.
18. Phillips-Wren, G., Hahn, E., Forgionne, G., 2004. A multiple
criteria framework for the evaluation of decision support
systems. Omega 32 (4), 323–332.
19. Phillips-Wren, G., Jain, L. (Eds.), 2005. Intelligent Decision
Support Systems in Agent-Mediated Environments. IOS Press,
Amsterdam.
20. Phillips-Wren, G., Mora, M., Forgionne, G., Garrido, L.,
Gupta, J.N.D., 2006a. Multi-criteria evaluation of intelligent
decision making support systems. In: Gupta, J.N.D.,
Forgionne, G., Mora, M. (Eds.), Intelligent Decision-Making
Support Systems (i-DMSS): Foundations, Applications and
Challenges. Springer, pp. 3–24.
21. Phillips-Wren, G., Mora, M., Forgionne, G., Gupta, J.N.D.,
2006b. Evaluation of decision-making support systems
(DMSS): An inte-grated dmss and AI approach. In: Adam, F.,
Humphreys, P. (Eds.), Creativity and Innovation in Decision
Making and Decision Support (Proceedings of CIDMDS
2006), London, UK, 29 June–1 July.
22. G. Phillips – Wern , M. Mora, G.A. Forgionne, J.N.D. Gupta, "
An integrative evaluation framework for intelligent decision,
European journal of Operational Research 195 (2009) 642-
652.
23. C.S. Pierce, in : C Hartehome, P. Weiss(Eds), collected
papers, vols. 1-8 , Harvard University Press, Cambridge,
1931-1958.
24. Pospelov D.A., Osipov G.S. Prikladnaja semiotika
(Applied semiotics) // Novosti iskusstvennogo intelekta
(Artificial Intelligence News), Moscow, 1999, N1. - P. 9-
35.
25. B. B. Rieger: Meaning Acquisition by SCIPS. In: Ayyub (ed):
ISUMA-NAFIPS-95, Los Alamitos, CA (IEEE) Computer
Society Press), 1995, pp. 390–395.
26. B. B. Rieger: Situations, Language Games, and SCIPS.
Modeling semiotic cognitive information processing
systems.In: Meystel/Nerode (eds): Architectures for Semiotic
Modeling and Situation Analysis in Large Complex Systems,
Bala Cynwyd, PA (AdRem), 1995, pp. 130–138.
27. Rosenthal-Sabroux, C., Zarate´, P., 1997. Artificial
intelligence tools for decision support systems. European
Journal of Operational Research 103, 275–276.
28. Saaty, T.L., 1977. A scaling method for priorities in
hierarchical structures. Journal of Mathematical Psychology,
234–281.
29. Saaty, T.L., 1986. How to make a decision: The analytic
hierarchy process. Interfaces 24 (6), 19–43.
30. Saaty, T., Vargas, L., 1994. Decision Making in Economic,
Political, Social and Technological Environments with The
Analytic Hierarchy Process. RWS Publications, Pittsburgh,
PA.
31. R. Stamper, Research issues in information systems:
semantics, in R. J. Boland, R.A. Hirschheim (Eds), critical
issues in information system research, Wiley, Chichester,
1987.
32. Vagin V.N., Yeremeyev A.P. Nekotorye bazovye principy
postroenija intellektual'nyh sistem podderzhki prinjatija
reshenij real'nogo vremeni (Some base principles of
construction of intellectual real time decision support
systems Izv. RAN: teorija i sistemy upravlenija
(Proceedings of the Russian Academy of Sciences: the
theory and control systems), 2001, N6. p. 114-123.
33. Wind, Y., Saaty, T.L., 1980. Marketing applications of the
analytic hierarchy process. Management Sciences 26 (7), 641–
658.
34. Xiao Fang, Ram Rachamadugu , "Polices for knowledge
refreshing in databases", www.elsevier.com/locate/omega.
35. A.P. Yeremeyev O. korrektnosti "produkcijnoj modeli
prinjatija reshenij na osnove tablic reshenij (About a
correctness of the production model of decision-making
on the basis of decision tables) Izv. RAN. Avtomatika i
telemekhanika (Proceedings of the Russian Academy of
Sciences: Automatics and telemechanics), 2001, N.10 p.
78-90.
36. Yeremeyev A.P., Denisenko L.S. Obrabotka
nedoopredelennoj informacii v sisteme podderzhki
prinjatija reshenij real'nogo vremeni primenitel'no k
operativnoj sluzhbe elektrostancij (Processing of not
predetermined information in real time decision support
systems with reference to operative service of power
stations) // Izv. RAN. Energetika (Proceedings of the
Russian Academy of Sciences: Power Engineering), 2002,
N.2. - P. 32-43.
37. A.P. Yeremeyev, Troickiy V.V. Modeli predstavlenija
vremennyh zavisimostej v intellektual'nyh sistemah
podderzhki prinjatija reshenij (Models of representation of
the time dependences in intellectual decision support
systems) Izv. RAN. Teorija i sistemy upravlenija
(Proceedings of the Russian Academy of Sciences: the
theory and control systems), 2003, N.5, p.75-88.
38. J. Zhang, Pu, P., 2006. Performance evaluation of consumer
decision support systems. International Journal of E-Business
Research 2 (3), 38–45.

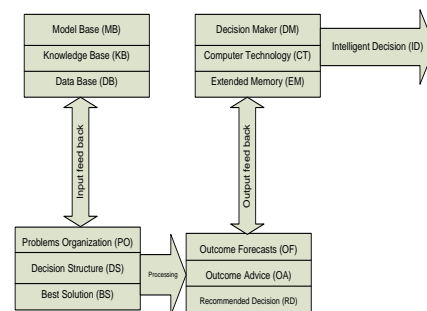


Figure 1 Intelligent decision support systems structure

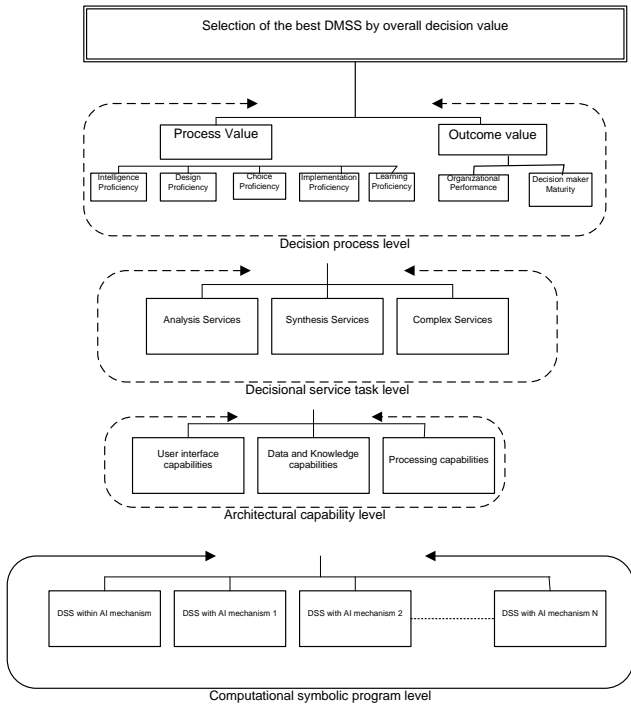


Figure 2 AHP model for IDSS evaluation

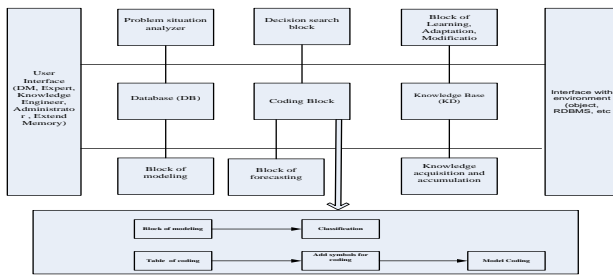


Figure 3 Suggested Architecture for Secured IDSS (SIDSS)

Study of Advances in Real-Time Simulation of Hybrid Electric Vehicles

Sachin kumar suryanshee, Mr.Arun Pachori,
Department of electrical engineering
Jabalpur engineering college Jabalpur (M.P) INDIA
E-mail Sachinsuryanshee@yahoo.in

Abstract: This paper starts with the basics of electric vehicle technology and introduces design principles of series hybrid electric vehicle. A series hybrid electric vehicle power train design study has been presented and a previously developed MATLAB/Simulink model has been used to simulate the designed vehicle in two drive cycles using a soft hybrid energy management strategy. Also performance simulations have been conducted for all electric drive mode and results have been compared with the measurements taken from an experimental vehicle.

Keywords: electric vehicle, Hybrid electric vehicle, Series hybrid electric vehicle ,modeling, simulation, electric vehicle design

1. Introduction

It is known that electric vehicle (EV) technology has been gaining importance at both military and commercial vehicle systems for the last decades. Despite they have higher cost, their higher energy efficiency, lower emissions, regenerative braking and silent mode drive capabilities are major advantages over conventional vehicles. Better performance of electric traction, suitability for future weapon systems, stealth mode, silent watch and reduced signature are some of the reasons for the growing interest on combat electric vehicles. This paper introduces electric vehicle technologies and basic design principals of series hybrid architecture. Performance of the designed vehicle is obtained using previously developed simulation environment.

2. Electric Vehicle Configurations

Basically electric vehicle configurations can be classified into three groups. They are,

- All electric vehicle
- Series hybrid electric vehicle
- Parallel hybrid electric vehicle

As shown in Figure 1, electric energy storage systems such as battery, flywheel and super

capacitor can be used as power supply system in all electric vehicles. In this configuration, the Range of the vehicle is limited by the stored electrical energy. Today, this is the most important drawback of EVs as a result of the weight of energy storage systems Any vehicle having two or more different type of power sources or drive system is called hybrid vehicle. Series hybrid electric vehicle (SHEV) is hybridization of power supply system.

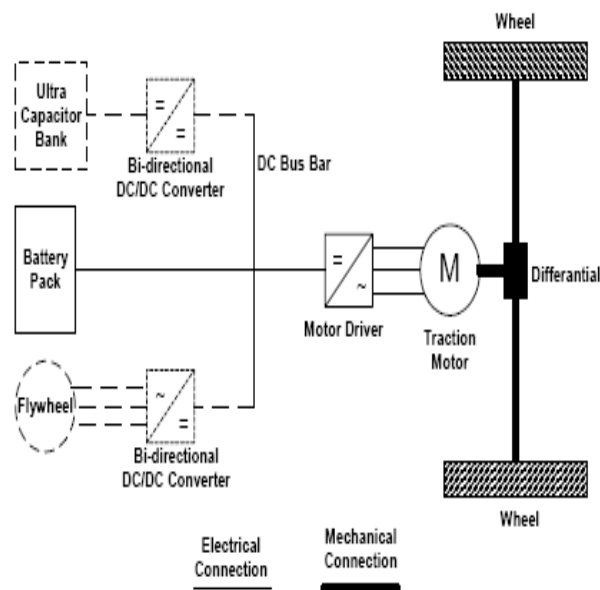


Figure 1. All Electric Vehicle Power train

At series hybrid configuration, an ICE-generator set is placed additional to all electric system (Figure 2). The generator set (genset) may act as electrical energy storage system state of charge (SOC) controller or as the main power unit. When used as main power supply, it covers average power demands and the energy storage system supplies peak loads. During deceleration or low power drive, energy storage system is charged by regenerative braking or genset.

Parallel hybrid electric vehicle (PHEV) is the hybridization of drive system. In parallel hybrid electric vehicles, both ICE and electrical machine can propel the

vehicle. For example, at low speeds, electric machine drives the car to use energy more efficiently. For better performance at long distance travels, ICE operates for traction. Electric machine can also act as a generator to charge energy storage systems if torque demand can be supplied by only ICE. One of the parallel-hybrid configurations is shown in Figure 3.

Beyond these two different hybrid configurations, as in Toyota Prius example, some other drive concepts have

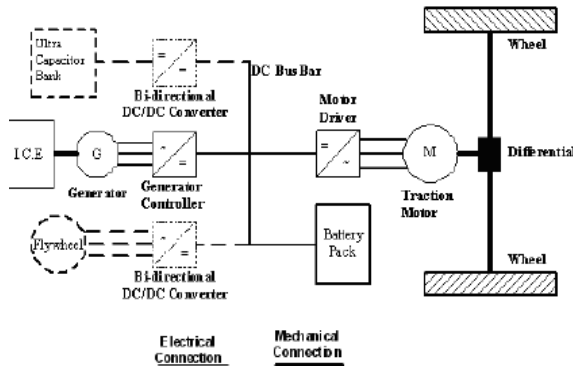


Figure 2. Series Hybrid Electric Vehicle Powertrain

Been studied like dual hybrid vehicles, which have properties of both SHEV and PHEV

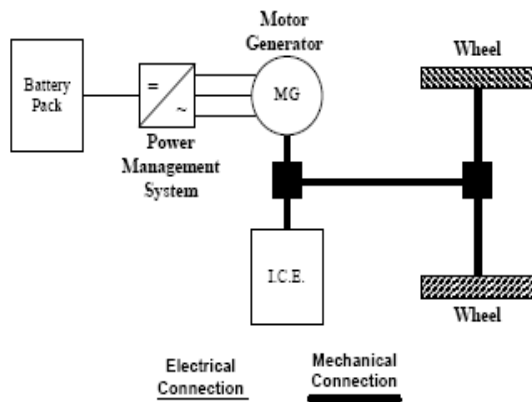


Figure 3. Parallel Hybrid Electric Vehicle Power train

3. Technologic Trends On Subsystems

Power system of an HEV consists of energy production unit, energy storage system(s), electric machine(s) and related power electronic circuits.

Present studies accept AC induction and permanent magnet synchronous machines as possible alternatives for electric drive applications. Switch reluctance motors may

be another interesting alternative. DC machines are not preferred anymore, as they need brush and commutator maintenance.

Having robust construction, low production cost and low maintenance requirement, induction machine has been common selection for experimental electric vehicles. Modern control techniques like field-oriented control properly provide necessary speed control function for induction machines.

Permanent magnet synchronous machines have high efficiency, low-weight, high power density and high speed possibilities. So far, in all commercial hybrid electric vehicles, permanent magnet motors have been used due to certain advantages defined above.

Fuel cells are the most promising technology for power generation, which convert the chemical energy of a fuel directly to electrical energy. When they take hydrogen as fuel, they produce only heat and steam as emission. Although there are several types of fuel cells, proton exchange membrane fuel cell (PEMFC) seems the most promising choice for electric vehicle applications as it has relatively small dimensions and low operation temperature (50-100°C). However, fuel cells have some serious disadvantages like high manufacturing, hydrogen production and storage costs. For today, usage of fuel cells as energy production unit in commercial hybrid vehicles is not yet feasible.

On the other hand, conventional energy production method, electric generator, takes mechanical energy and produces electrical energy. Mechanical power source can be a diesel engine, a gas fuel engine or a gas turbine. AC induction or permanent magnet machines can be used as generator.

As mentioned before, the worst problem in front of electric vehicle technology is the energy storage. Until now, lead acid (PbA) batteries have sustained the leadership despite their low specific energy and power ratios. Reliability and ability to withstand rough conditions are some reasons for that. In addition, advanced PbA battery technologies like spiral wound AGM (absorbent glass mat) or VRLA (Valve regulated lead acid) batteries are able to provide higher specific power ratios up to 400 W/kg.

	Specific Energy Ratio (Wh/kg)	Specific Power Ratio (W/kg)	Cycle Life at %80 DOD (cycles)
PbA	30-45	150-400	300-550
NiMH	60-70	150-1300	800-1300
Li-Ion	90-130	250-1400	500-1200

Table 1. Battery Comparison

Studies on both nickel metal hydride (NiMH) and lithium ion (Li-ion) batteries are going on. Higher specific energy and power ratios, better charge absorption during regenerative braking and longer cycle life are some of the promised attributes of these advanced technologies. In commercial hybrid vehicles like Toyota Prius and Honda Insight, NiMH batteries are used.

Although some data has been given in Table 1, it should be noted that these parameters are only for indicative purposes since the data may have wide variations among different battery manufacturers and these data always change with the advancement of battery technology.

As flywheels, mechanical energy storage system, have high specific power ratio but low specific energy ratio, they cannot be used as main energy storage devices but can be a good alternative for short-term peak power demands. Another alternative for supplying peak loads is super capacitor. However, these systems also bring cost and reliability issues.

4. Powertrain Design

As mentioned before, power system of SHEV consists of traction motor, power generation unit, energy storage system and associated power electronics. Choosing proper traction motor and its supply system are the main issues. Series hybrid electric vehicle power system design starts with choosing the traction motor. Vehicle specifications like weight, friction force, desired nominal velocity, acceleration and gradability affect this choice. The selected electric motor has to overcome several forces, which are wheel friction force (Ft), air friction force (Fr), slope friction force (Fe) and force due to vehicle inertia (Fa). The required power from traction motor is $P_m = F_{total} \cdot V$.

Forces acting on to the vehicle;

$$F_t = c_t \cdot m \cdot g \cdot \cos \alpha \quad (1)$$

$$F_e = m \cdot g \cdot \sin \alpha \quad (2)$$

$$F_r = 0.5 \cdot c_r \cdot \delta \cdot A_f \cdot V^2 \quad (3)$$

$$F_a = m \cdot dV/dt \quad (4)$$

F_t , F_e and F_r are used to calculate the continuous power requirement from electric motor and F_r is used to determine the additional power for acceleration. Total power that should be transferred to the wheels is obtained by multiplying the total force (F_{tot}) and vehicle speed (8). Total wheel torque (T_{tot}) is the product of force and wheel radius (9).

$$F_p = F_t + F_e + F_r \quad (5)$$

$$F_d = F_a \quad (6)$$

$$F_{tot} = F_d + F_p \quad (7)$$

$$P_{tot} = F_{tot} \cdot V \quad (8)$$

$$T_{tot} = F_{tot} \cdot r \quad (9)$$

Reduction gear ratio and mechanical efficiency is also considered while selecting the traction motor specification. If direct drive is applied, transmission efficiency (μ_{teff}) may also be omitted.

$$P_m = P_{tot} / \mu_{teff} \quad (10)$$

$$w_{tire} = V / r \quad (11)$$

$$w_{rotor} = w_{tire} \cdot GR \quad (12)$$

$$T_m = P_m / w_r \quad (13)$$

Gradability target is one of the most important parameter for the nominal torque requirement and maximum motor speed is one of the parameters determining the maximum vehicle speed.

Power supply system (Pss) of hybrid electric vehicle is configured considering energy management strategy and power requirement of consumers such as traction motor, cooling system (Pc) and auxiliaries (Paux).

$$P_{ss} = P_{genset} + P_{bat} = P_m / \mu_{meff} + P_c + P_{aux} \quad (14)$$

Load sharing between power generation and energy storage devices and total energy that should be stored are influenced by the energy management system. Longer silent drive range requires higher energy storage. The other way is that generator set covers the average load and energy storage device supply short-term peak power.

Table 2. Formula constants

Variable	Description	Unit	Variable	Description	Unit
m	Total vehicle mass	kg	α	Road slope	°
c_t	Wheel friction coefficient	-	c_r	Air friction coefficient	-
g	Gravity 9.81	m/s ²	δ	Air density	kg/m ³
A_f	Vehicle frontal area	m ²	V	Vehicle speed	m/s

The proposed design activity is conducted on the series hybrid electric vehicle power train architecture shown in figure 4.

Table 3. Vehicle constant

Constant	Value	Constant	Value
ct	0.01	cw	0.3
δ (km/m ³)	1.17	Af (m ²)	3.1
nt	% 90	rw (m)	0.325
mt (kg)	1600	g (m/s ²)	9.81
GR	5.7276		

Using equations given in (5) to (12), forces acting on the vehicle were modeled. Using the vehicle constants in Table 3, to be able to determine the specifications of traction motor, input signals, speed and slope, were applied to the model and following power-speed, torque-speed graphics were calculated.

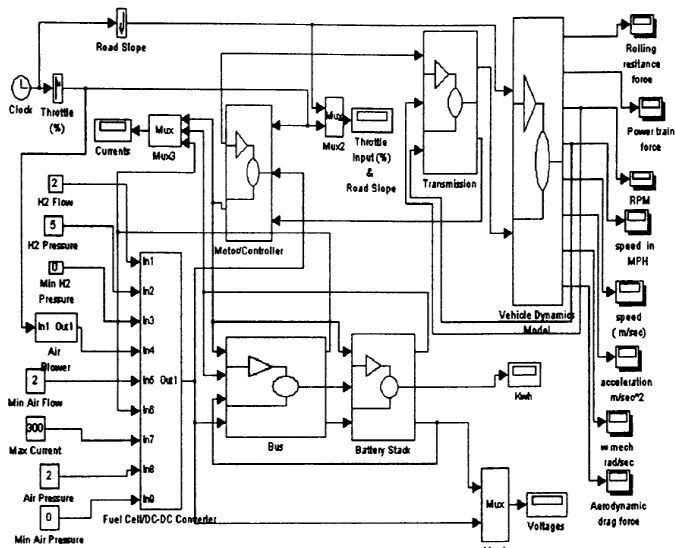


Figure 4. Simulink Model of a series HEV with Fuel CeU as the APU unit.

0 to 90 km/h acceleration in 25 seconds at straight road input signals were applied to the model and power-motor angular velocity and torque-motor angular velocity values were plotted. Pivme and Psabit show power requirement during acceleration and requirement at constant speed drive respectively.



Figure 5. Maximum speed power requirements

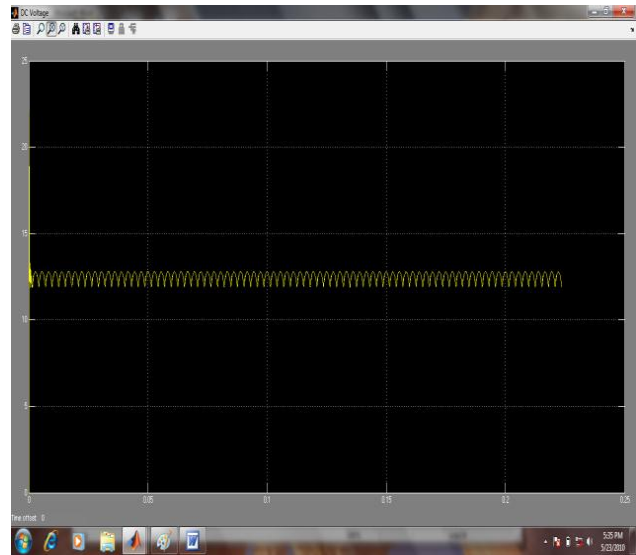


Figure 6. Maximum speed load moment

The same model was run for acceleration to 50 km/h in 25 seconds but this time at %10 slope and outputs were plotted in figure 7 and figure 8.

The graphs show that for this vehicle an electric motor having 250 Nm maximum torque at low speed (0-2000 rpm) and 50-60 kW maximum power between 2000-5000 rpm is needed.

For SHEV, the electric power supply system design is a more complex problem than the drive system. Selected power management method influences the parts of it. There are two basic energy management strategy called like soft hybrid and power assist. In the first one, battery pack may act as the main supplier and genset may be used as the battery SOC controller. In the other method genset is the main supplier for average power requirement and batteries are used for supplying peak loads.

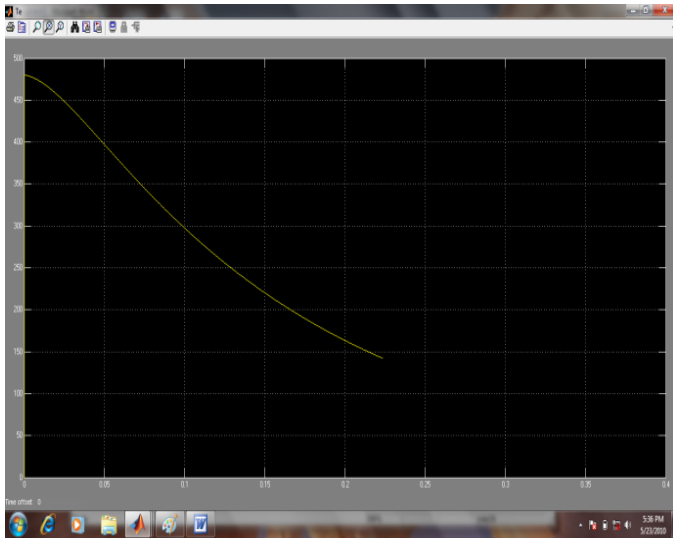


Figure 7. % 10 slope power requirement

For this study, power management is selected as soft hybrid. Battery pack can supply all required power from traction motor. When battery pack state of charge decreases under a certain limit, generator set starts to supply electrical energy to recharge the battery pack.



Figure 8. % 10 slope load moment

In this context, battery pack should be able to supply at least 60 kW. 300 V DC bus bar, which seems suitable for power electronics at this power level, can be composed by 25 x 12 V lead acid batteries in series.

Capacity of the battery pack should be selected considering both maximum current supply capability and silent mode drive range of the vehicle. Selecting silent drive range as 150 km at 50 km/h constant speed, analyzing the calculations conducted before, brings the

requirement of 18 kWh total stored energy. Division of maximum discharge power to total capacity results in 3.3 C discharge rate, which may be allowable for advanced lead acid batteries.

Generator set should be able to supply recharging power when the SOC decreases predefined ranges and also supply power for the traction at moderate speed drive. Selecting 9 kW nominal recharge power as the half of total capacity and taking 5 kW power requirement for 50 km/h drive into consideration, genset is selected at 15 kW power level.

In the soft hybrid energy management strategy, battery pack SOC level is divided into various operating modes, such as given in figure 9

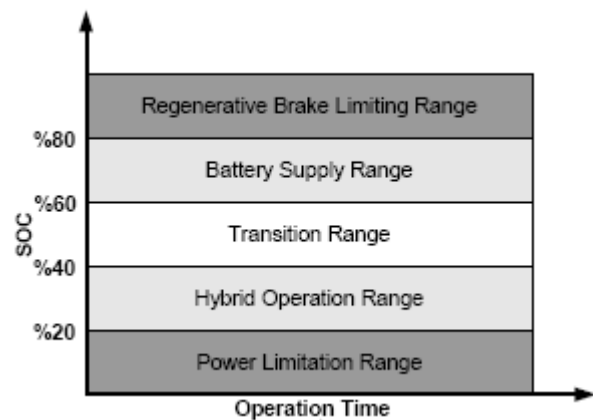


Figure 9. Soc operation bands

One wants to operate in the %20 - %80 SOC range. This is mainly because of the high internal resistance of battery pack behind those limits. So, above %80 SOC, regenerative braking is simply omitted or limited to avoid over charge and below %20 SOC, power limitation for traction is applied to avoid deep discharge. When SOC is in the %60-%80 range, required power is supplied only by battery pack till SOC decreases to %40. At this point, genset starts to supply power to the dc bus until SOC reaches %60. While driving down hill battery pack may increase upto %80.

Designed vehicle has been simulated using previously developed MATLAB/Simulink environment. The first drive cycle applied is Urban Dynamometer Drive Schedule (UDDS) of USA federal test procedure FTP-75. The second one is the US06 highway cycle of USA test procedure.

In the urban drive cycle, only battery pack supplied the dc bus and genset didn't operate. At the end of 23 minute drive cycle, vehicle traveled 12 km/h distance and the battery SOC decreased %8. Seeing that the traction motor power was usually under 20 kW, 15 kW generator set

would be sufficient to keep battery SOC between the required operation band.

In the high-speed drive cycle, after 25 km drive, battery pack SOC decreases around %35 and dc bus bar voltage floats between 250 V - 360 V.

5. Comparison Of Performance Simulations And Real Case

For the pure electric mode drive, acceleration and constant speed power consumption simulations were conducted and the results were compared with an experimental electric vehicle. Simulations were conducted under 35 kW power supply limitation to compare the results with the real case.that vehicle accelerates to 60 km/h in 13 seconds in the simulation mode. Experimental vehicle accelerated to the same speed in 14.4 seconds

The simulation showed that power consumption at 50 km/h is 5 kW. In the real case power consumption was measured as 5.9 kW. Reasons for the difference between the simulation and real case are thought to be the uncertain vehicle parameters and efficiency values used in the model.

6. Conclusion

In this paper, basics of hybrid electric vehicle technology have been presented. Also a series hybrid electric power train design study has been conducted.

The designed vehicle has been simulated using previously developed MATLAB/Simulink model. The developed model can be optimized to introduce new topologies as well as building up energy management strategies. Resulted simulation is thought to be an optimized full series parallel hybrid electric vehicle By obtaining proper results for two different characterized

Driving cycles, this claim is supported It is thought to be a good information source for automotive Producers and users, and even governmental or civil organizations to realize the potentials of hybrid electrical Vehicle benefits.

References

1. C. Gökce, Modeling and Simulation of a Series-Parallel Hybrid Electrical Vehicle, Master Thesis, Istanbul Technical University, Jun. 2005.
2. Van den Bossche, P., Van Mierlo, J., Maggetto, G., "Energy Sources for Hybrid electric Vehicles: Comparative Evaluation of The State of The Art", AECV conference, Noordwijkerhout/The NETHERLANDS, 2002

3. TUR, O., Simulation of Hybrid Electric Vehicle Power System, MSc. Thesis, Istanbul Technical University, 2004
4. TÜBTAK MRC Energy Institute, ELT – 2 Paralel Hibrit Elektrikli Araç, Research Project submitted to TOFAS, Dec. 2004
5. Barsali, S., Pasquali, M., Pede, G., 2002, Definition of an Energy Management Technique for Series Hybrid Vehicles, Electric Vehicle Symposium 19, Busan/KOREA, October 2002
6. Lee, Y.K., Jung, J.H., 2002, Optimal Power Distribution Strategy for Series Hybrid Electric Vehicle Using Diesel Engine and Battery System, Electric Vehicle Symposium 19, Busan/KOREA, October
7. C. Dufour, J. Bélanger, T. Ishikawa, K. Uemura, "Real-Time Simulation of Fuel Cell Hybrid Electric Vehicles", Proceedings of the 2004 Global Powertrain Congress, September 28-30, 2004, Dearborn, MI USA.

Author Biographies



Sachin Kumar Suryvanshee was born in Balaghat (M.P.), India in 25 February, 1984. He received the B.E. degree from Rajiv Gandhi Proudhyogiki Vishwavidyalaya Bhopal, india in 2007 in electrical engineering and M.E. degree from Rajiv Gandhi Proudhyogiki Vishwavidyalaya Bhopal, india in 2010 in high voltage and power system.

His research focuses the area of modeling and simulation of vehicles, power systems, high voltage and electro magnetic fields.



Mr. Arun Pachori was born in Jabalpur (M.P.), India in 11 July, 1961. He received the B.E. degree from Rani Durgavati Vishwavidyalaya Jabalpur, india in 1983 in electrical engineering and M.E. degree from Rajiv Gandhi Proudhyogiki Vishwavidyalaya ,Jabalpur , india in 2003 in high voltage engineering .

He is currently Reader in High Voltage in the department of Electrical Engineering at Jabalpur Engineering College, Jabalpur (M.P.) India . He has 24 years Teaching experience. His research focuses the area of high voltage , power transformer and lightning over voltage .

Performance of Reversible Image Watermarking Algorithm using Histogram Shifting under Noisy Conditions

S. Kurshid Jinna¹ Dr. L. Ganesan²

¹Professor, Dept of Computer Science & Engineering
PET Engineering College Vallioor, Tirunelveli, India

²Professor, Dept of Computer Science & Engineering
A.C College of Engineering & Technology, Karaikudi, India

kurshidjinna@gmail.com, csehod@gmail.com

Abstract: Image transfer leads to addition of unavoidable noise to the watermarked image. Sometimes malicious attacks also introduces noise to the image which replaces parts of the image deliberately by another piece. A method of lossless data hiding in images using integer wavelet transform and histogram shifting for gray scale images is proposed. The method shifts part of the histogram, to create space for embedding the watermark information bits. The method embeds watermark while maintaining the visual quality well. The method is completely reversible. The original image and the watermark data can be recovered without any loss if noise is not introduced. Noise interference imparts difficulty in exact retrieval of the embedded watermark and the original image. Effect of noise while extracting the watermark and reconstructing the original image is studied.

Keywords: Noise, Data Hiding, Histogram shifting, reversible watermarking, attacks, watermark retrieval.

1. Introduction

The reversible watermarking algorithms are developed from the time it was suggested by its pioneers. Fridrich et al, Jun Tian and Ni et al are pioneers in the field.

Ni et al. [1] proposed an image lossless data hiding algorithm using pairs of zero-points and peak-points, in which the part of an image histogram is shifted to embed data. lossless data embedding algorithm based on the histogram shifting in spatial domain is proposed. J.Fridrich and M. Goljan suggested general methodologies for lossless embedding that can be applied to images as well as any other digital objects. The concept of lossless data embedding can be used as a powerful tool to achieve a variety of necessary tasks, including lossless authentication using fragile watermarks [2].

J. Tian calculates the differences of neighboring pixel values, and selects some difference values for the difference expansion (DE) for reversible data embedding as suitable pairs for data embedding. Pairs which do not affect the algorithm for lossless embedding and extraction are used and is indicated with the help of location map [3].

Xuan et al.[4] proposed the lossless embedding using the integer wavelet transform (IWT) and histogram medication using a threshold point for embedding limit. G. Xuan and Y. Q. Shi proposed a histogram shifting method for image lossless data hiding in integer wavelet transform domain. This algorithm hides data into wavelet coefficients of high frequency subbands. It shifts part of the histogram of high frequency wavelet subbands and embeds data by using the created histogram zero-point [5]. Chrysochos et al's scheme of reversible watermarking presents a method resistant to geometrical attacks [6].

Fallahpour M, Sedaaghi M proposes relocation of zeroes and peaks of the histogram of the image blocks of the original image to embed data in the spatial domain. Image is divided into varying number of blocks as required and the performance is analysed. [7]

Xianting Zenga, Lingdi Ping and Zhuo Li proposed scheme based on the difference histogram shifting to make space for data hiding. Differences of adjacent pixels are calculated by using different scan paths. Due to the fact that the grayscale values of adjacent pixels are close to each other, the various-directional adjacent pixel difference histogram contains a large number of points with equal values; data hiding space is obtained [8].

As a progress to this research domain, effect of noise on images is studied. Noise destroys the ability to retrieve the embedded watermark thereby resulting in loss of valuable information [9]. Various noises are introduced and the effect is examined by varying the intensities of each one of them. Noise removal from images plays a vital role in the success of various applications. These applications include optical character recognition, content-based image retrieval and hand-written recognition systems [10].

The formation causes of speckle noise in the reconstructed image and its characteristics are studied. Some conditions aggravate the speckle noise and is a kind of multiplicative noise in the reconstructed image [11].

2. Integer-To-Integer Wavelet Transforms

In conventional wavelet transform reversibility is not achieved due to the floating point wavelet coefficients we get after transformation. When we take the inverse transform the original pixel values will get altered.

When we transform an image block consisting of integer-valued pixels into wavelet domain using a floating-point wavelet transform and the values of the wavelet coefficients are changed during watermark embedding, the corresponding watermarked image block will not have integer values. When we truncate the floating point values of the pixels, it may result in loss of information and reversibility is lost. The original image cannot be reconstructed from the watermarked image.

In conventional wavelet transform done as a floating-point transform followed by a truncation or rounding it is impossible to represent transform coefficients accurately. Information will be potentially lost through forward and inverse transforms.

In view of the above problems, an invertible integer-to-integer wavelet transform based on lifting is used in the proposed scheme. It maps integers to integers which are preserved in both forward and reverse transforms. There is no loss of information. Wavelet or subband decomposition associated with finite length filters is obtained by a finite number of primal and dual lifting followed by scaling.

3. Wavelet Histogram Shifting

Integer Wavelet transforms of the original image is taken. In the subband wavelet histogram

data is to be embedded. In the histogram the horizontal axis(X) represents the wavelet coefficients value and the vertical axis(Y) represents the number of occurrence of the coefficients value. The wavelet histogram normally exhibits a Laplacian distribution nature with a peak point and sloping on either side. Peak in wavelet histogram is usually at coefficient value '0'

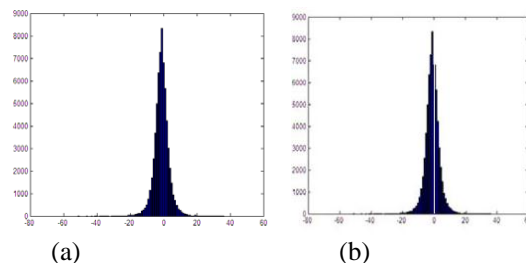
Embedding can be done on both the sides of the histogram to get the required embedding capacity.

Data embedding is done by modifying some of the coefficient values of the wavelet domain to it's neighboring value by shifting a portion of the histogram. This gives a good visual quality and thereby a better PSNR between original image and watermarked image.

To embed data we choose the peak point of the histogram and call it as P. Figure 1 shows a vacant point is created at Peak+1. This is done by shifting all points with value Peak+1 and above one position to the right. Now all the IWT coefficients are scanned and whenever a coefficient with value peak is encountered, '0' is embedded by leaving it as such and '1' is embedded by changing it's value to peak+1. This is repeated till all the points with value Peak are over. Then a new peak is created by shifting to the right and data is embedded as per the algorithm. We choose the peak point so that payload is maximized.

All the high frequency wavelet subbands can be utilized to get maximum capacity. The same process can be done on the left side of the histogram Peak to embed more watermark bits. A reverse algorithm is applied for extracting the watermark data.

After water mark bits are extracted, shifting is done towards the left each time after data extraction so that the original coefficient values are restored. This guarantees complete reversibility and the original image can be exactly reconstructed without loss.



(a) (b)
Figure 1 Illustration of wavelet Histogram, (a) Maximum point is at Peak, (b) Histogram with zero point created at peak +1

4. Proposed Method

4.1 Embedding Method

For the wavelet transformed image sub bands histogram is taken. Now we can start embedding using the following steps. For the selected sub band, set $P = \text{Peak}$ of the histogram coefficients.

Create a zero point at $P+1$ so that no point in the histogram has the value $P+1$. To create the zero point shift all coefficients with value $P+1$ and above to one position right. This makes $P+1$ as $P+2$, and the original $P+2$ to $P+3$ and so on.

1. Now positions P and $P+1$ are chosen to embed data.
2. Read the n watermark bits W_b where $0 < b < n-1$.
3. Check $W_b = 0$, then '0' is embedded in the coefficient with value P by leaving it unchanged as P .
4. Check $W_b = 1$, then '1' is embedded in the coefficient with value P by changing it to value $P+1$.
5. Point $P+1$ gets slowly filled up depending upon the number of W_b bits with value 1.
6. Go to histogram of the other sub bands to be marked and repeat the same process.
7. While to- be- embedded watermark bits are still remaining, set $P = \text{Peak} + 2$ and go to step 1. Otherwise stop.

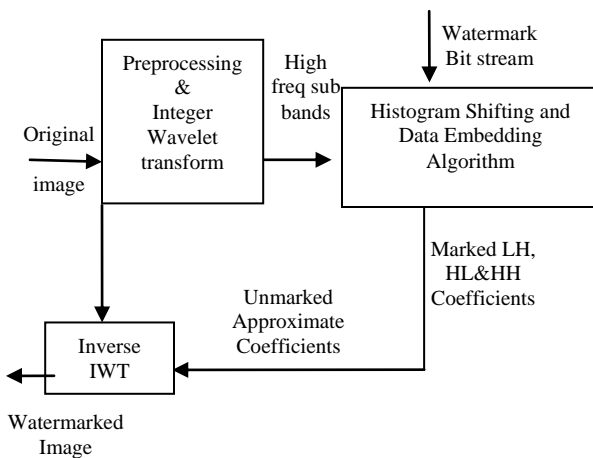


Figure 2 Embedding Method

Figure 2 Shows the original image is decomposed into it's sub bands using integer wavelet transform

After preprocessing IWT is used to ensure complete reversibility. The high frequency sub bands (horizontal, Vertical and Diagonal) are used for data embedding. Each sub band is used one after the other to meet the required embedding capacity. Watermark bits that forms the payload is embedded into these sub bands using the embedding algorithm. The low frequency unmarked approximate coefficients are then used along with the marked sub bands and Inverse IWT is taken to get the watermarked image.

4.2 Extraction Method

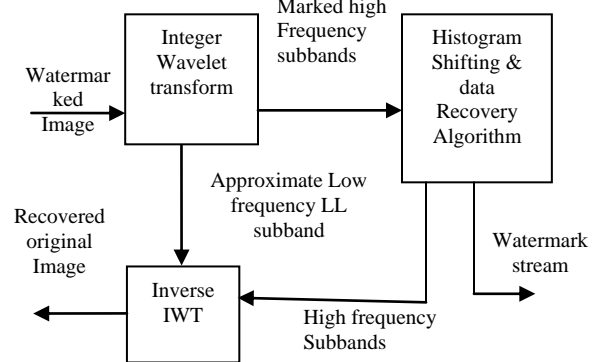


Figure 3 Extraction Method

The extraction method is shown in figure 3. Data extraction is the reverse process. Integer wavelet Transform is taken for the watermarked image. The watermarked high frequency sub bands are separated and using the Data extraction algorithm, the watermark bits are retrieved and the original sub bands are obtained. This is combined with the unmarked low frequency sub band to get the original image. This method is completely blind and reversible. Original image and the watermark data bits are obtained without any loss.

After wavelet decomposition of the watermarked image, histograms of the marked sub bands are taken. For the selected sub band, set $\text{Peak} = \text{Peak}$ of the histogram coefficients.

1. $P = \text{Peak}$. Read the coefficients with value P and $P+1$. Whenever a coefficient with value P is read ,extract watermark bit as $W_b = 0$ and leave P unaltered. Whenever a coefficient with value $P+1$ is read ,extract watermark bit as $W_b = 1$ and change $P+1$ to P .
2. Shift all the coefficients with value $P+2$ and above one position to the left.

3. Go to histogram of the other marked sub bands and repeat the same process.
4. Set $P = \text{Peak} + 1$.
5. While all watermark bits W_n are not extracted go to step 1. Otherwise stop.

5. Image Noise

Image noise is generally regarded as an undesirable by-product during image transfer. These unwanted fluctuations became known as "noise" and they interfere in the extraction of watermark embedded in an image for authentication and security.

Noise is of various types. They are generally distributed in the whole image or part of the image. The types of noise which we have used for testing the algorithm are Salt Pepper noise, Gaussian noise, Speckle noise and Poisson noise.

Another category of disturbance arises due to the fact that part(s) of the image is intentionally altered by replacing it by another part. In this case also the embedded watermark as well as the reconstructed original image get disturbed and are not retrieved exactly as the original one.

5.1 Salt and Pepper Noise

It is a form of noise typically seen on images. It represents itself as randomly occurring white and black pixels. An effective noise reduction method for this type of noise involves the use of a median filter. Salt and pepper noise creeps into images in situations where fast transients, such as faulty switching occur.

In salt and pepper noise, pixels in the image are very different in color or intensity from the surrounding pixels. The defining characteristic is that the value of a noisy pixel bears no relation to the intensity of surrounding pixels. Generally this type of noise will only affect a small number of image pixels. When viewed, the image contains dark and white dots, so it is termed salt and pepper noise.

5.2 Gaussian Noise

In Gaussian noise, each pixel in the image will be changed from its original value by a small amount. A histogram plot of the amount of distortion of a pixel value against the frequency, with which it occurs, shows a normal distribution

of noise. While other distributions are possible, the Gaussian distribution is usually a good model, due to the fact that the sums of different noises tend to approach a Gaussian distribution according to central limit theorem.

The standard model of amplifier noise is additive, Gaussian, independent at each pixel and independent of the signal intensity, caused primarily by Nyquist noise also called thermal noise, including that which comes from the reset noise of capacitors. Amplifier noise is a major part of the "read noise" of an image sensor, that is, of the constant noise level in dark areas of the image.

5.3 Speckle Noise

Speckle noise is a granular noise that inherently exists in and degrades the quality of the active synthetic aperture radar (SAR) images.

Speckle noise results from random fluctuations in the return signal from an object that is no bigger than a single image-processing element. It increases the mean grey level of a local area. Speckle noise in SAR is generally more serious, causing difficulties for image interpretation. It is caused by coherent processing of scattered signals received from multiple distributed targets.

Speckle noise in SAR is a multiplicative noise, and it is in direct proportion to the local grey level in any area. The signal and the noises are statistically independent of each other. The sample mean and variance of a single pixel are equal to the mean and variance of the local area surrounding that pixel.

5.4 Poisson Noise

Poisson noise or shot noise is a type of electronic noise. This occurs when number of electrons in an electronic circuit or photons in an optical device, is small enough to give rise to detectable statistical fluctuations in a measurement. It is important in telecommunications and image transmission. This type of noise follows a Poisson distribution, which is usually not very different from Gaussian noise.

In many cases, noise values at different pixels are modeled as being independent and identically distributed and hence uncorrelated.

Tests are conducted for both the cases where noise affects a portion of the image or the whole image. Experiments are conducted by introducing these noises at different amounts and area. Experiments are also conducted by replacing part of watermarked image by one or more icons. Different amount of noise is introduced on each image. The effect on the embedded watermark is studied by extracting the watermark from the noise affected image.

Results show how the embedded watermark gets affected by different noises.

6. Experimental Results and Discussion

6.1 Performance under normal condition

Experiments are conducted using different 512 X 512 gray scale images and different wavelets



Figure 4 Watermarked Image with payload (bpp) 0.4 (a)Sail Boat PSNR 37.05 dB, (b) Woman Dark Hair PSNR 45.03 dB, (c) Camera Man PSNR 48.65 dB, (d) Lena PSNR 37.89 dB, (e) Jet Plane PSNR 42.35 dB, (f) Lake PSNR 36.42 dB

Figure 4 shows image quality tested on different gray scale images after embedding around 1,00,000 bits.

Table 1 shows that cameraman image has a better embedding capacity than other images in the

experiment. It also shows it has a better visual quality as far as Peak signal to noise ratio is concerned.

Figure 5 shows image quality tested on cameraman gray scale image after embedding different payload bits.

Figure 6 shows the image quality tested for different images using integer wavelet transform for different payloads using cdf2.2 wavelet .The sailboat image though has higher quality for the same payload compared to Lena image using lower payload, the image quality quickly falls down as payload is increased.

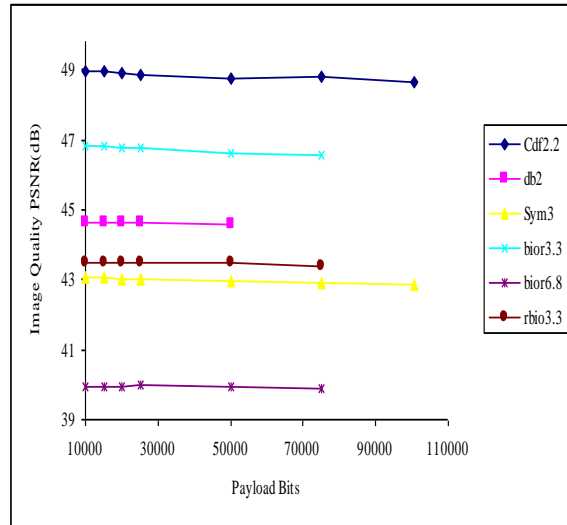


Figure 5 shows image quality tested on cameraman gray scale image for different payload bits.

Table 1 Image Quality Tested for Different Grayscale Images for each payload using Cdf2.2 wavelet

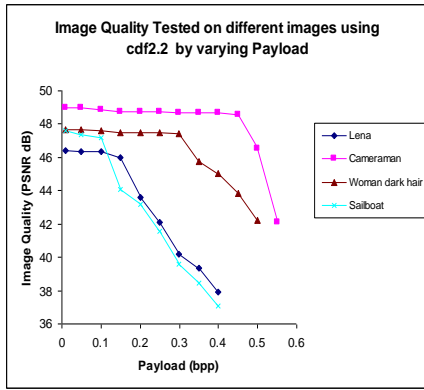


Figure 6 Image Quality Tested on Different Images Using cdf 2.2 by Various Payload (bpp)

Table II Comparison of performance of various wavelet families on Cameramen for different payload size

Peak signal to noise ratio exists while changing the wavelet family used for decomposing the original image for embedding. Also PSNR between the original image and the watermarked image varies depending on the image when using the same wavelet for decomposition.

Table III Image Quality Tested for Different Grayscale Images for fixed payload of 25000 bits

Payload Bits	cdf2.2 PSNR (dB)	db2 PSNR (dB)	sym3 PSNR (dB)	bior3.3 PSNR (dB)	bior6.8 PSNR (dB)	rbio3.3 PSNR (dB)
10000	48.96	44.63	43.05	46.85	39.93	43.50
15129	48.95	44.62	43.04	46.81	39.93	43.49
20164	48.91	44.61	43.03	46.78	39.93	43.48
25281	48.86	44.60	43.02	46.76	39.99	43.47
50176	48.75	44.56	42.93	46.61	39.94	43.46
75076	48.78	xxx	42.90	46.55	39.86	43.36
100489	48.67	xxx	42.83	xxx	xxx	xxx

Wavelets	Lena PSNR (dB)	Cameraman PSNR (dB)	Woman Dark Hair. PSNR (dB)	Sail Boat PSNR (dB)
db1	47.56	48.32	49.37	48.85
cdf2.2	46.34	46.58	48.87	47.59
bior3.3	45.37	45.53	46.75	46.38
sym2	44.62	44.98	44.60	44.83
db3	42.35	42.51	42.14	42.44
sym3	41.12	41.16	43.02	42.51
rbio3.3	41.09	41.32	43.51	42.87
rbio6.8	40.45	40.79	40.48	40.66
bior6.8	39.88	40.40	39.99	40.24

Table II shows image quality tested for different payloads on the same image using different wavelets. Cdf2.2 performs better than other wavelets for the same payload. Image quality quickly changes when different wavelets are used. Performance in embedding measured using peak signal to noise ratio shows that bior6.8 has the minimum quality. The embedding capacity also varies when using different wavelets using different wavelets when the image is decomposed using db2 embedding stops in about 50,000 bits whereas cdf2.2 continues to embed over one lakhs bits. The same is illustrated in the graph.

Payload (bpp)	Lena PSNR (dB)	Camera man PSNR (dB)	Woman Dark Hair. PSNR (dB)	Sail Boat PSNR (dB)
0.1	46.3338	48.8519	47.5824	47.2015
0.15	45.9927	48.7678	47.5095	44.0572
0.2	43.6043	48.7591	47.5146	43.1822
0.25	42.1267	48.7421	47.4691	41.5522
0.3	40.1725	48.7014	47.4321	39.5865
0.35	39.3304	48.6990	45.7253	38.4327
0.4	37.8965	48.6552	45.0346	37.0477
0.45	xxx	48.5825	43.8402	xxx
0.5	xxx	46.5234	42.2365	xxx
0.55	xxx	42.1261	xxx	xxx

Experiments were conducted on various 512x512 grayscale images to study the performance of various wavelets on the embedding algorithm. For a fixed payload of 25,000 bits embedded and tested, db1 performs best as shown in the table III. Bior6.8 has the minimum quality. A variation of about 10db in

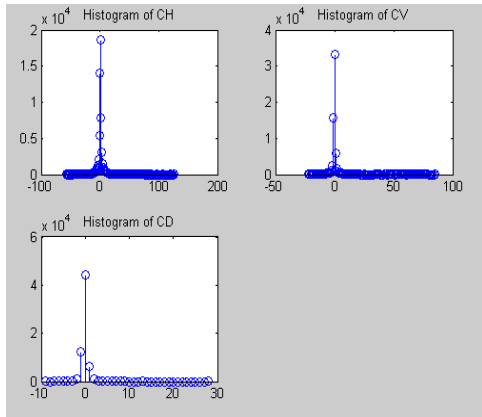


Figure 8 Histogram of Cameraman Image after IWT

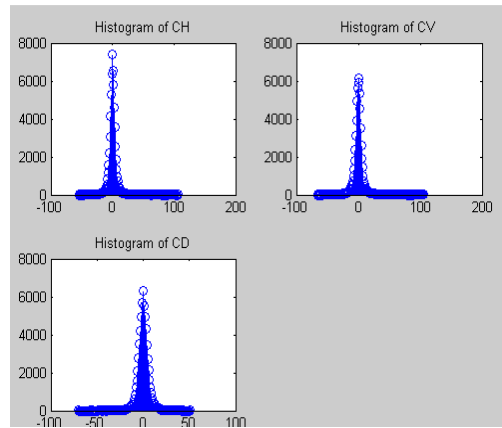


Figure 11 Histogram of Lena Image after IWT

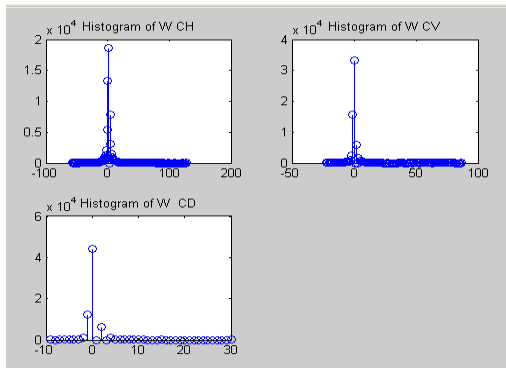


Figure 9 Histogram of Watermarked Cameraman Image

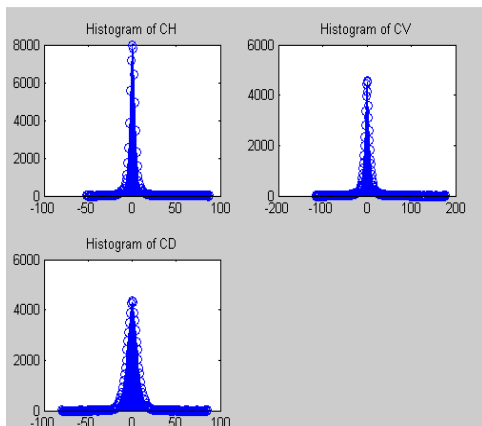


Figure 10 Histogram of Sailboat Image after IWT

Histogram of wavelet transformed cameraman image shows more number of coefficient values at peak point compared to Lena image and sailboat. This influences the embedding capacity. Cameraman image has higher embedding capacity compared to Lena or sailboat image. This is illustrated in figure 8, 10 and 11. Figure 9 shows the watermarked cameraman image wavelet histogram. This shows the shifted positions of the histogram points due to shifting and embedding.

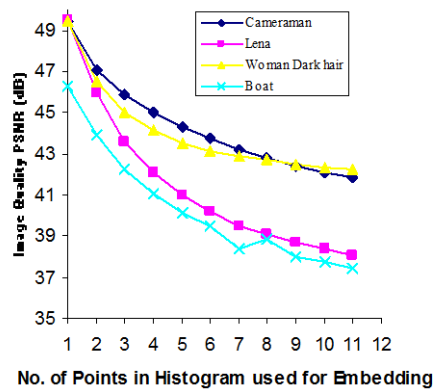


Figure 12 Image Quality Tested by using different number of embedding points in histogram

Image quality is tested using different number of embedding points in histogram to embed the watermark data. Each coefficient value can embed watermark bits equal to the number of occurrence of that point in the wavelet histogram. Figure 12 shows image quality decreases as we use more and more points in the histogram for embedding data. With lesser payload fewer points are used and we get more image quality for the watermarked images.

6.2 Performance under Noisy Conditions

6.2.1 Noise affecting part of an Image



Figure 13 10% Salt Pepper Noise in image part

Fig 13 shows salt pepper noise affecting a corner of the Elaine image. The recovered image also shows visual artifacts because of the noise. The extracted watermark is also not retrieved as the original embedded watermark.



(a) Recovered Image (b) Difference Image

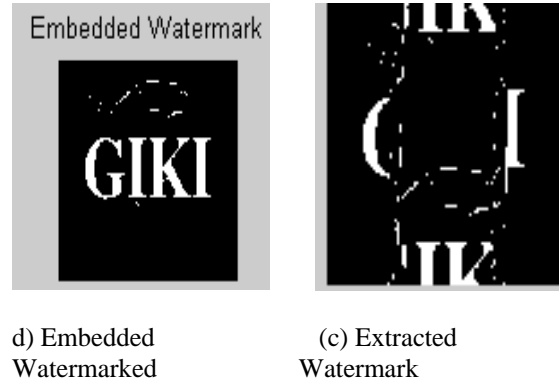


Figure 14 10% Gaussian Noise in image part

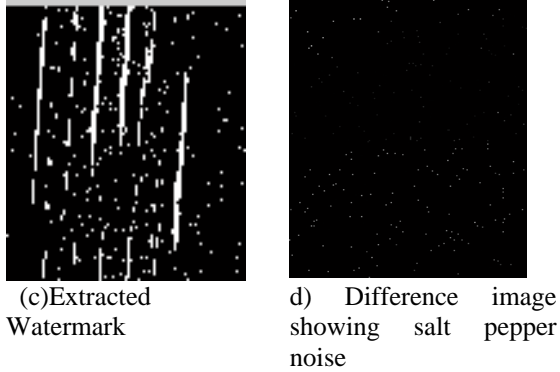
Fig 14 illustrates similar effect because of added Gaussian noise in part of the Elaine image

6.2.2 Whole Image

Noise affects either part of the image or the whole image. Fig 15 shows how salt pepper noise throughout the boat image disturbs the recovered image as well as the extracted watermark.

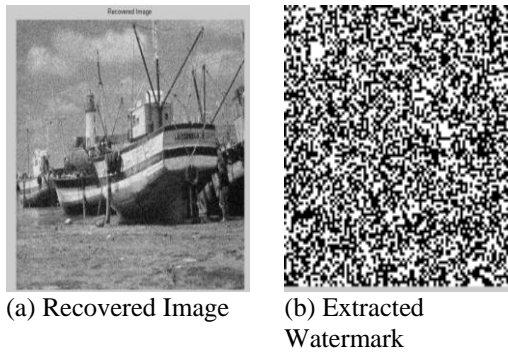


(a) Original image (b) Recovered Image



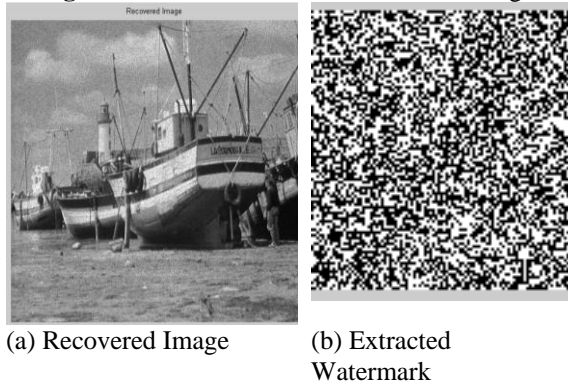
(c) Extracted Watermark
 (d) Difference image showing salt pepper noise

Figure 15 1% Salt Pepper Noise in whole image



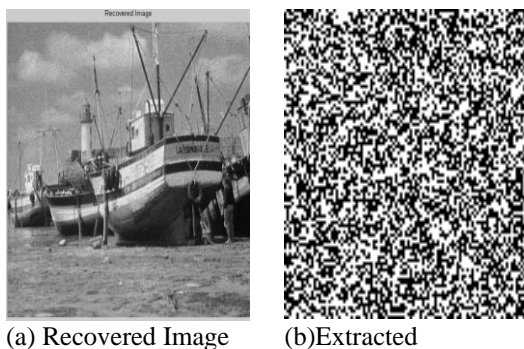
(a) Recovered Image
 (b) Extracted Watermark

Figure 16 1% Gaussian Noise in whole image



(a) Recovered Image
 (b) Extracted Watermark

Figure 17 1% Speckle noise in whole image



(a) Recovered Image
 (b) Extracted Watermark

Watermark

Figure 18 1% Poisson noise in whole image

Figure 16-18 shows similar effect caused by Gaussian noise, speckle noise and poisson noise in the whole boat image. The extracted watermark is also distorted.

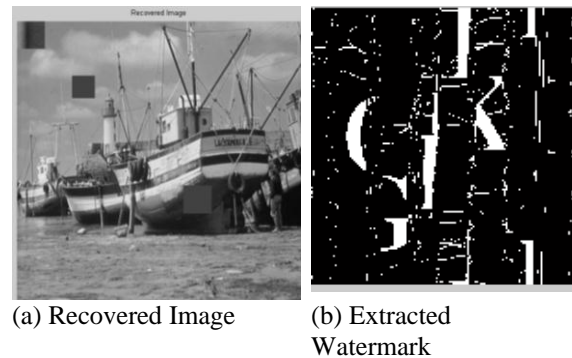
6.2.3 Image Tampered using Icons

Sometimes part of the watermarked image is replaced intentionally with a different part. This also disturbs the reconstructed original image and the extracted watermark. Fig 19 shows cameraman image tampered in two places. One at the corner of the image and the other near the hand of the cameraman. The reconstructed original image illustrates the same. Fig 20 shows boat image tampered with three icons. The extracted watermark is affected more compared to the case of two icons.



(a) Recovered Image
 (b) Extracted Watermark

Figure 19 Cameraman Image tampered with two icons



(a) Recovered Image
 (b) Extracted Watermark

Figure 20 Boat Image tampered with three icons

7. Conclusion

Reversible image watermarking using histogram shifting method was done and tested using different images. Embedding capacity not only varies from image to image, it also varies for various wavelets. The wavelet histogram is used for embedding as it has a Laplacian like distribution and embedding can be done on both sides of the histogram to embed more data. More image quality is achieved for the same payload compared to other reversible watermarking methods. Images with more number of points on the wavelet histogram peak can embed more data. This is a blind watermarking method. Original image and the embedded data are extracted exactly without any loss because our method is completely reversible if noise is not added to the watermarked image. In the presence of noise the embedded watermark is not exactly recovered. The effect of various noise interference in extracting the watermark and reconstructing the original image is studied.

8. References

1. Z. Ni, Y. Q. Shi, N. Ansari, and S. Wei, "Reversible data Hiding," in ISCAS Proceedings of the 2003 International Symposium on Circuits and Systems, vol. 2, pp. II-912-II-915, Thailand, May 2003
2. J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," in SPIE Proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, vol. 4675, pp. 572-583, San Jose, Jan 2002
3. J. Tian, "Reversible data embedding using a difference expansion," in IEEE Transactions on Circuits Systems and Video Technology, vol. 13, no. 8, pp. 890-896, Aug 2003
4. G. Xuan, Y. Q. Shi, C. Yang, Y. Zheng, D. Zou, P. Chai, "Lossless data hiding using integer wavelet transform and threshold embedding technique," IEEE International Conference on Multimedia and Expo (ICME05), Amsterdam, Netherlands, July 6-8, 2005.
5. G. Xuan, Y. Q. Shi, Q. Yao, Z. Ni, C. Yang, J. Gao, P. Chai, "Lossless data hiding using histogram

shifting method based on integer wavelets," International Workshop on Digital Watermarking (IWDW 2006), Nov. 8 - 10, 2006, Jeju Island, Korea.

6. Chrysochos E., Fotopoulos V., Skodras A., Xenos M., "Reversible Image Watermarking Based on Histogram Modification", 11th Panhellenic Conference on Informatics with international participation (PCI 2007), Vol. B, pp. 93-104, 18-20 May 2007, Patras, Greece.

7. Fallahpour M, Sedaaghi M. "High Capacity lossless data hiding based on histogram modification". IEICE Electronic Express, Vol. 4, No. 7, April 10, 2007 page 205-210

8. Xianting Zeng, Lingdi Ping, Zhuo Li "Lossless Data Hiding Scheme Using Adjacent Pixel Difference Based on Scan Path" Journal of Multimedia, Vol. 4, No. 3, June 2009

9. Rafael C. Gonzalez, Richard E. Woods (2007) Digital Image Processing. Pearson Prentice Hall. ISBN 013168728X.

10. "Novel Adaptive Filtering for Salt-and-Pepper Noise Removal from Binary Document Images", Springer Volume 3212/2004 pages 191-199

11. "Reduction of speckle noise in the reconstructed image of digital holography" Xiao-ou Cai Optik - International Journal for Light and Electron Optics Volume 121, Issue 4, February 2010, Pages 394-399

Author Biographies



S. Kurshid Jinna Completed her B.E in Electronics and Communication Engineering from Thiagarajar College of Engineering, Madurai, in 1985 and M.E(Hons) in Computer Engineering from VJTI, University of Mumbai and doing Ph.D in faculty of information and communication in Anna University, Chennai. She is currently working as Professor & head of the department, Computer Science and Engineering in PET Engineering College, Vallioor, India.



Dr. L.Ganesan completed his B.E in Electronics and Communication Engineering from Thiagarajar College of Engineering, Madurai and M.E in Computer Science and Engineering from Government College of Technology, Coimbatore. He completed his Ph.D from Indian Institute of Technology, Kharagpur in the area image processing. He has authored more than fifty publications in reputed International Journals. His area of interest includes image processing, multimedia and compressions. He is currently working as head of the department of Computer science and engineering, A.C. College of Engg. And Technology, Karaikudi, India

A Novel Image Compression Algorithm based on Discrete Wavelet Transform with Block Truncation Coding

Shyam Lal¹, Mahesh Chandra² & Gopal Krishna Upadhyay³

¹Department of E & C Engineering, Moradabad Institute of Technology, Moradabad-244001(U.P)-India

²Department of E & C Engineering, Birla Institute of Technology, Mesra-Ranchi(Jharkhand)

³Director, SSITM, Kasganj, Moradabad (U.P.) –India

[shyam_rao24@rediffmail.com, shrotriya@bitmesra.ac.in, gkupadhyay2003@yahoo.com]

Abstract- This paper presents a novel image compression algorithm for gray scale image. Digital images contain large amount of information that need evolving effective techniques for storing and transmitting the ever increasing volumes of data. Image compression addresses the problem by reducing the amount of data required to represent a digital image. Image compression is achieved by removing data redundancy while preserving information content. In this paper a simplified and more efficient image compression algorithm is described and implemented. This paper proposed an efficient image compression algorithm which is based on block truncation coding (BTC). Simulation & Experimental results on benchmark test images demonstrate that the new approach attains competitive image compression performance, compared with state-of-the-art image compression algorithms.

Keywords: Image compression, DWT, SPIHT, Haar Transform and Block Truncation Coding

1. Introduction

The basic idea behind this method of compression is to treat a digital image as an array of numbers i.e., a matrix. Each image consists of a fairly large number of little squares called pixels (picture elements). The matrix corresponding to a digital image assigns a whole number to each pixel. For example, in the case of a 256x256 pixel gray scale image, the image is stored as a 256x256 matrix, with each element of the matrix being a whole number ranging from 0 (for black) to 255 (for white). Image compression is used to minimize the amount of memory needed to represent an image. Images often require a large number of bits to represent them, and if the image needs to be transmitted or stored, it is impractical to do so without somehow reducing the number of bits. The problem of transmitting or storing an image affects all of us daily. TV and fax machines are both examples of image transmission, and digital video players and web pictures of Catherine Zeta-Jones are examples of image storage [1-3].

Image compression is a technique used to reduce the storage and transmission costs. The existing techniques used for compressing image files are broadly classified into two categories, namely lossless and lossy compression techniques. In lossy compression techniques, the original digital image is usually transformed through an invertible linear transform into another domain, where it is highly de-correlated by the transform. This de-correlation concentrates the important image information into a more compact form. The transformed coefficients are then quantized yielding bit-

streams containing long stretches of zeros. Such bit-streams can be coded efficiently to remove the redundancy and store it into a compressed file. The decompression reverses this process to produce the recovered image [1-3].

Discrete Cosine Transform (DCT) is a powerful mathematical tool that took its place in many compression standards such as JPEG and MPEG. In the most general form, DCT can be expressed as matrix multiplication. The 2-D discrete cosine transform (DCT) is an invertible linear transform and is widely used in many practical image compression systems because of its compression performance and computational efficiency [1-3]. DCT converts data (image pixels) into sets of frequencies. The first frequencies in the set are the most meaningful; the latter, the least. The least meaningful frequencies can be stripped away based on allowable resolution loss. DCT-based image compression relies on two techniques to reduce data required to represent the image. The first is quantization of the image's DCT coefficients; the second is entropy coding of the quantized coefficients [4]. Quantization is the process of reducing the number of possible values of a quantity, thereby reducing the number of bits needed to represent it. Quantization is a lossy process and implies in a reduction of the color information associated with each pixel in the image [4 -7].

Haar Wavelet Transform (HWT): The 1D Haar Transform can be easily extended to 2D. In the 2D case, we operate on an input matrix instead of an input vector. To transform the input matrix, we first apply the 1D Haar transform on each row. We take the resultant matrix, and then apply the 1D Haar transform on each column. This gives us the final transformed matrix. The 2D Haar transform is used extensively in efficient image compression, both lossless and lossy [8].

The JPEG compression technique divides an image into 8x8 blocks and assigns a matrix to each block. One can use some linear algebra techniques to maximize compression of the image and maintain a suitable level of detail. JPEG (Joint Photographic Experts Group) is an international compression standard for continuous-tone still image, both grayscale and color. This standard is designed to support a wide variety of applications for continuous-tone images. Because of the distinct requirement for each of the applications, the JPEG standard has two basic compression methods. The DCT-based method is specified for lossy compression, and the predictive method is specified for lossless compression. A simple lossy technique called baseline, which is a DCT-based methods, has been widely

used today and is sufficient for a large number of applications. In this paper, we will simply introduce the JPEG standard and focuses on the baseline method [9-10].

The Set Partition in Hierarchical Tree (SPHT) algorithm is unique in that it does not directly transmit the contents of the sets, the pixel values, or the pixel coordinates. What it does transmit is the decisions made in each step of the progression of the trees that define the structure of the image. Because only decisions are being transmitted, the pixel value is defined by what points the decisions are made and their outcomes, while the coordinates of the pixels are defined by which tree and what part of that tree the decision is being made on. The advantage to this is that the decoder can have an identical algorithm to be able to identify with each of the decisions and create identical sets along with the encoder [11].

Wavelet coding is proving to be a very effective technique for image compression, giving significantly better results than the JPEG standard algorithm with comparable computational efficiency. The standard steps in such compression are to perform the Discrete Wavelet Transform (DWT), quantize the resulting wavelet coefficients (either uniformly or with a human visual system weighting scheme), and losslessly encode the quantized coefficients. These coefficients are usually encoded in raster-scan order, although common variations are to encode each sub-block in a raster-scan order separately or to perform vector quantization within the various sub-blocks [12-16].

In this paper we present an efficient image compression algorithm by discrete wavelet transform (DWT) block truncation coding (BTC) method. This algorithm gives better image compression performance.

The rest of paper is structured as follows. Section II briefly review the concept of discrete wavelet transform (DWT). Section-III presents the proposed image compression algorithm in detail. Section-IV presents the simulation & experimental results and section-V concludes the paper.

2. Discrete Wavelet Transform

Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale. They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Wavelets were developed independently in the field of mathematics, quantum physics, electrical engineering, and seismic geology. Interchanges between these fields during the last 20 years have led to many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction. This paper introduces wavelets to the interested technical person outside of the digital signal processing field. Some researchers describe the history of wavelets beginning with Fourier, compare wavelet transforms with Fourier transforms, state properties and other special aspects of wavelets, and finish with some interesting applications such as image compression, musical tones, and de-noising noisy data. The fundamental idea behind wavelets is to analyze according to scale. Indeed, some researchers in the wavelet

field feel that, by using wavelets, one is adopting a whole new mindset or perspective in processing data.

The Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. This idea is not new. Approximation using superposition of functions has existed since the early 1800's, when Joseph Fourier discovered that he could superpose sines and cosines to represent other functions. However, in wavelet analysis, the scale that we use to look at data plays a special role. Wavelet algorithms process data at different scales or resolutions. If we look at a signal with a large "window" we would notice gross features. Similarly, if we look at a signal with a small "window" we would notice small features. The result in wavelet analysis is to see both the forest and the trees, so to speak. This makes wavelets interesting and useful [7][12-16].

Mathematically the Discrete wavelet transform pair for one dimensional can be defined as

$$W_\phi(j_0, k) = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} f(x) \tilde{\phi}_{j_0, k}(x) \quad (1)$$

$$W_\psi(j, k) = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} f(x) \tilde{\psi}_{j, k}(x) \quad (2)$$

for $j \geq j_0$ and

$$f(x) = \frac{1}{\sqrt{M}} \sum_k W_\phi(j_0, k) \phi_{j_0, k}(x) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_\psi(j, k) \psi_{j, k}(x) \quad (3)$$

Where $f(x)$, $\phi_{j_0, k}(x)$, and $\psi_{j, k}(x)$ are functions of discrete variable $x = 0, 1, 2, \dots$,

In two dimensions, a two-dimensional scaling function, $\phi(x, y)$, and three two-dimensional wavelet $\psi^H(x, y)$, $\psi^V(x, y)$ and $\psi^D(x, y)$, are required. Each is the product of a one-dimensional scaling function ϕ and corresponding wavelet ψ .

$$\phi(x, y) = \phi(x)\phi(y) \quad (4)$$

$$\psi^H(x, y) = \psi(x)\phi(y) \quad (5)$$

$$\psi^V(x, y) = \phi(y)\psi(x) \quad (6)$$

$$\psi^D(x, y) = \psi(x)\psi(y) \quad (7)$$

where ψ^H measures variations along columns (like horizontal edges), ψ^V responds to variations along rows (like vertical edges), and ψ^D corresponds to variations along diagonals. The two-dimensional DWT can be implemented using digital filters and down samplers and it is shown in the Figure 1 & 2, respectively.

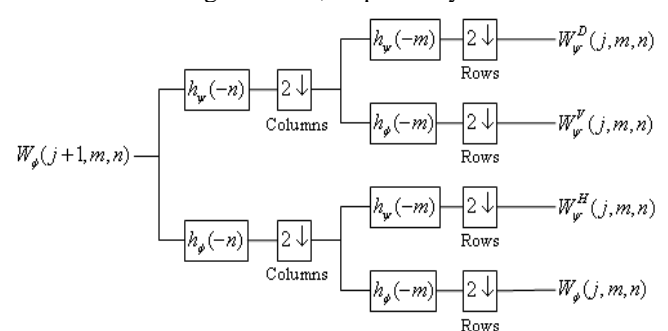


Fig. 1. The two-dimensional DWT—the analysis filter

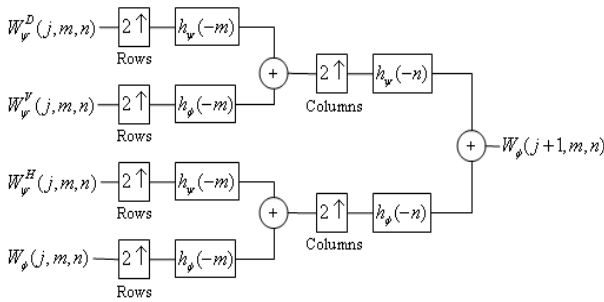


Fig. 2. The two-dimensional DWT—the synthesis filter

III. PROPOSED ALGORITHM

The block diagram of proposed algorithm is given below

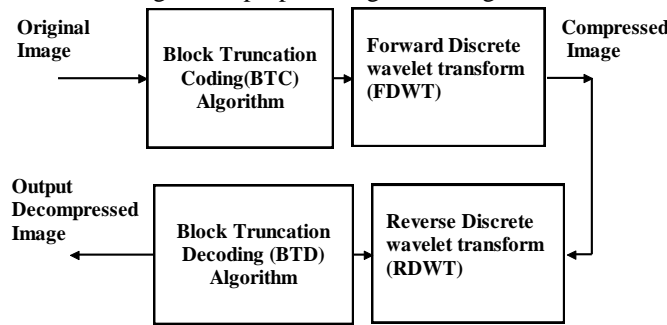


Fig.3. Block diagram of proposed image compression and decompression algorithm

A. Block Truncation Coding & Decoding Algorithm:

The main goal is to transmit or to store only data related to the statistics of the image blocks of the original image using a coding algorithm. A decoding algorithm is also needed to display the image using only the statistics of the blocks. Block Truncation Coding Algorithm (BTC) is simple, fast and essentially robust against transmission errors [17-18].

For the study presented here the image (NXN) will be divided into 4X4 pixel blocks, and the quantizer will have 2 levels. After dividing the image into MXM blocks ($M = 4$ for our examples), the blocks are coded individually, each into a two level signal. The levels for each block are chosen such that the first two sample moments are preserved. Let $M = N^2$ and let X_1, X_2, \dots, X_M be the values of the pixels in a block of the original picture.

Then the first and second sample moments and the sample variance are respectively

$$\bar{X} = \frac{1}{M} \sum_{i=1}^M X_i \quad (8)$$

$$\overline{X^2} = \frac{1}{M} \sum_{i=1}^M X_i^2 \quad (9)$$

$$\sigma^2 = \overline{X^2} - (\bar{X})^2 \quad (10)$$

As with the design of any one bit quantizer, we find a threshold X_{th} , and two output levels, A and B , such that

$$\left. \begin{array}{l} \text{if } X_i \geq X_{th} \quad \text{output} = B \\ \text{if } X_i < X_{th} \quad \text{output} = A \end{array} \right\} \text{ for } i = 1, 2, \dots, M \quad \text{For our}$$

first quantizer, we set $X_{th} = \bar{X}$ and the output levels A and B are found by solving the following equations:

Let Q =number of X_i 's greater than $X_{th} (= \bar{X})$ then to preserve \bar{X} and $\overline{X^2}$

$$M \bar{X} = (M - Q)A + QB \quad (13)$$

and

$$M \overline{X^2} = (M - Q)A^2 + QB^2 \quad (14)$$

Solving Equation (13) & (14) for A and B

$$A = \bar{X} - \bar{\sigma} \sqrt{\frac{Q}{M - Q}} \quad (15)$$

$$B = \bar{X} + \bar{\sigma} \sqrt{\frac{M - Q}{Q}} \quad (16)$$

$$\text{Where } \bar{\sigma} = \sqrt{\overline{X^2} - (\bar{X})^2}$$

Each block is then described by the values of \bar{X} , $\bar{\sigma}$ and an $N \times N$ bit plane consisting of 1's and 0's indicating whether pixels are above or below X_{th} . Assuming 8-bits each to \bar{X} and $\bar{\sigma}$ results in a data rate of 2 bits/pixel.

a. The coding Algorithm Implementation procedure:

1. The original image ($N \times N$) is broken down into small blocks of size ($M \times M$) pixels with $M \ll N$; usually $M = 4$.
2. For each block, the mean value \bar{X} and the mean square value $\overline{X^2}$ are computed, as well as $\bar{\sigma}$.
3. Each pixel value of the block is then compared to a threshold value.

$$\left. \begin{array}{l} \text{if } X_i \geq X_{th} \quad \text{output} = B \\ \text{if } X_i < X_{th} \quad \text{output} = A \end{array} \right\} \text{ for } i = 1, 2, \dots, M \quad (17)$$

The block is replaced with a block of A's and B's. The image is therefore converted to an image with pixels values equal to A or B.

4. The code table is

$$T(i, j) = \begin{cases} 1 & \text{if } A \\ 0 & \text{if } B \end{cases} \quad (18)$$

5. The mean value \bar{X} and $\bar{\sigma}$ are then coded with 8 bits.
6. For each block, \bar{X} , $\bar{\sigma}$, and $T(i, j)$ are transmitted.
7. Repeat for each block of the original image.

b. The decoding Algorithm Implementation procedure:

1. The received data, \bar{X} , $\bar{\sigma}$, and $T(i, j)$ for each block are converted to real numbers.
2. Then, for each block the values A and B are computed as

$$A = \bar{X} - \bar{\sigma} \sqrt{\frac{Q}{M - Q}} \quad (19)$$

$$B = \bar{X} + \sigma \sqrt{\frac{M - Q}{Q}} \quad (20)$$

The block is then reconstructed as

$$Y(i,j) = A.I(i,j) + (B-A).T(i,j) \quad (21)$$

for $i=1,2,3,4$ & $j=1,2,3,4$.

Where $I(i,j)$ is the identity block.

3. Repeat for each block received.
4. The last step is to rebuild the whole image from the reconstructed blocks.

B. DWT Image Compression Algorithm :

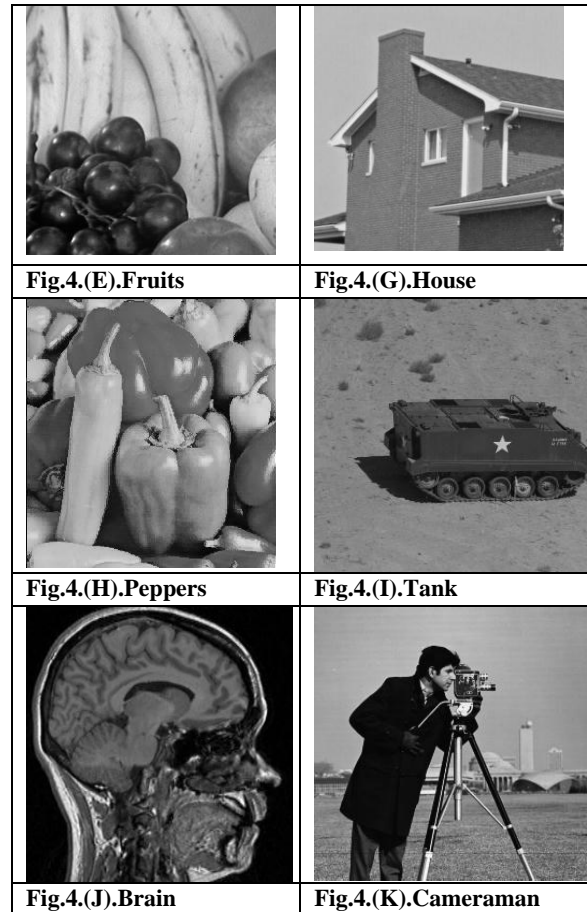
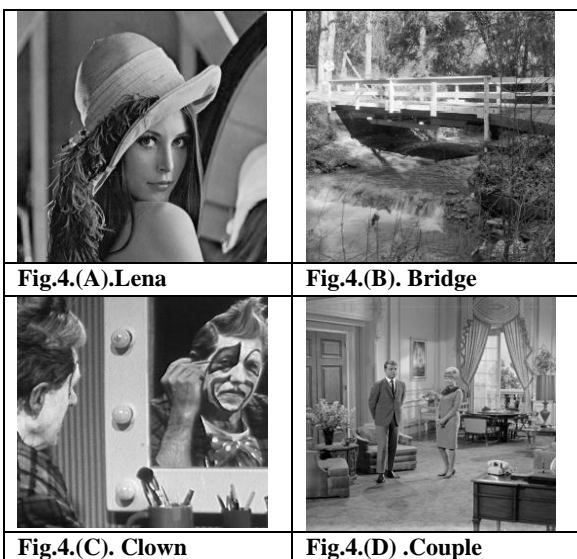
The idea behind this method is that further it performs image compression very efficiently and gives higher compression ratio and better image qualities.

The following implementation steps have been made for the image compression using discrete wavelet transform, which is based on 2-D-wavelet.

- Reading coded image from the output of BTC
- Decomposition of images using wavelets for the level N.
- Selecting and assigning a wavelet for compression.
- Generating threshold coefficients using Birge-Massart strategy.
- Performing the image compression using wavelets.
- Computing and displaying the results such as compressed image, retained energy and Zero coefficients.
- Decompression of the image based on the wavelet decomposition structure.
- Plotting the reconstructed image.
- Computing and displaying the size of original image, compressed image and decompressed image.

3. Simulation & Experimental Results

The 8-bit images of dimensions $M \times M$ ($= 256 \times 256$) pixels is used for simulation. For testing the performance of proposed image compression algorithm, the following test images shown in Fig.4 were used for experiments.



The proposed image compression algorithm can be viewed as completion and extension of discrete wavelet transform. We compared performance of proposed algorithm with five state-of-the-art image compression: Haar wavelet transform, JPEG, SPIHT and DWT. Test gray –scale image(size:256X256) used in our experiments. We Evaluated and compared performance of different image compression algorithms by using three measures: Compression ratio(CR), Peak signal to noise ratio(PSNR),and mean squared error(MSE). Although compression ratio measure the rate of image compression and PNSR can measure the intensity difference between two images, It is well known that it may fail to describe the visual perception quality of image. The superiority of proposed algorithm is demonstrated by conducting two experiments. Compression ratio (CR) is defined in equation (22) and peak signal to noise ratio (PSNR) in dB as defined in equation (23) and Mean squared error (MSE) are the metrics used to compare the performance of proposed image compression algorithm with existing algorithms.

1. Compression Ratio: The image compression ratio is defined as

$$CR = \frac{\text{Number of bits in Original image}}{\text{Number of bits in Compressed image}} \quad (22)$$

2. Peak signal to noise ratio (PSNR): The PSNR between the filtered output image $y(i, j)$ and the original image $s(i, j)$ of dimensions $M \times M$ pixels is defined as:

$$PSNR = 10 * \log_{10} \left(\frac{MAX_I^2}{\sqrt{MSE}} \right) \quad (23)$$

Where MAX_I the maximum pixel value of the image and MSE is is mean squared error and it is defined as

$$MSE = \frac{\sum_i \sum_j [y(i, j) - s(i, j)]^2}{M1 \times M2} \quad (24)$$

A. Experiment 1:

Table-I gives the image compression performance in terms of compression ratio. It can be seen from Table-I that proposed algorithm gives higher compression ratio as compared to existing lossy compression techniques. Table-II gives the image compression performance in terms of peak signal to noise ratio (PSNR) in dB. It can be seen from Table-II that proposed algorithm gives higher PSNR as compared to HWT & DWT and lower as compared to JPEG & SPIHT. Table -III gives the image compression

performance in terms of mean squared error (MSE).It can be seen from Table-III that proposed algorithm gives lower MSE as compared to HWT & DWT and higher as compared to JPEG and SPIHT.

So we have to trade-off between compression ratio and Peak to signal to noise ratio (PSNR).

B. Experiment 2:

To visualize the image quality of decompressed image of proposed image compression algorithm is compared with existing image compression techniques such as HWT, JPEG, SPIHT and DWT. Fig.5,6,7,8,9,10,11,12,13, and 14 gives visual appearance of Lena, Bridge, Fruits, Clown, Couple, House, Peppers, Tank, Brain and Cameraman by HWT, JPEG, SPIHT, DWT and proposed algorithm.

Table-I. Compression Ratio performance of various image compression algorithms

Method/Image	HWT	JPEG	SPIHT	DWT	Proposed
Lena(256X256)	5.73	35.17	53.20	50.73	56.33
Bridge(256X256)	3.00	52.50	53.80	59.00	59.18
Clown(256X256)	5.94	52.80	54.43	54.60	55.27
Fruit(256X256)	11.48	38.57	49.73	50.79	50.94
Couple(256X256)	5.94	40.80	44.11	52.35	53.48
House(256X256)	9.20	40.53	40.53	40.53	40.63
Peppers(256X256)	6.43	31.9	36.34	39.40	42.50
Tank(256X256)	8.80	25.10	30.21	33.04	33.81
Brain(256X256)	5.04	12.38	8.75	36.65	37.16
Cameraman(256X256)	4.28	14.74	30.65	34.85	39.13

Table-II. PSNR performance of various image compression algorithms

Method/Image	HWT	JPEG	SPIHT	DWT	Proposed
Lena(256X256)	24.26	29.36	35.79	26.43	27.73
Bridge(256X256)	23.00	26.16	28.32	23.56	24.90
Clown(256X256)	24.26	30.00	35.02	25.31	26.77
Fruit(256X256)	23.81	32.71	38.57	29.61	31.16
Couple(256X256)	23.81	28.70	33.36	25.84	27.19
House(256X256)	24.87	32.42	38.66	30.74	32.34
Peppers(256X256)	24.39	29.60	36.03	26.49	27.52
Tank(256X256)	26.30	32.76	40.35	33.67	34.86
Brain(256X256)	23.63	30.00	35.98	25.45	26.86
Cameraman(256X256)	25.33	27.86	34.70	26.04	27.17

Table-III. MSE performance of various image compression algorithm

Method/Image	HWT	JPEG	SPIHT	DWT	Proposed
Lena(256X256)	219.38	75.30	17.12	147.85	56.33
Bridge(256X256)	325.65	157.16	95.52	286.20	210.36
Clown(256X256)	243.78	64.87	20.43	191.23	136.61

Fruit(256X256)	270.46	34.78	9.03	70.17	49.74
Couple(256X256)	270.76	87.12	29.90	169.25	124.17
House(256X256)	211.76	37.19	8.85	54.33	37.86
Peppers(256X256)	236.25	31.90	16.21	145.62	115.05
Tank(256X256)	152.37	34.37	5.98	27.92	21.19
Brain(256X256)	281.71	62.30	16.38	155.20	132.96
Cameraman(256X256)	190.66	106.23	21.99	161.73	124.47

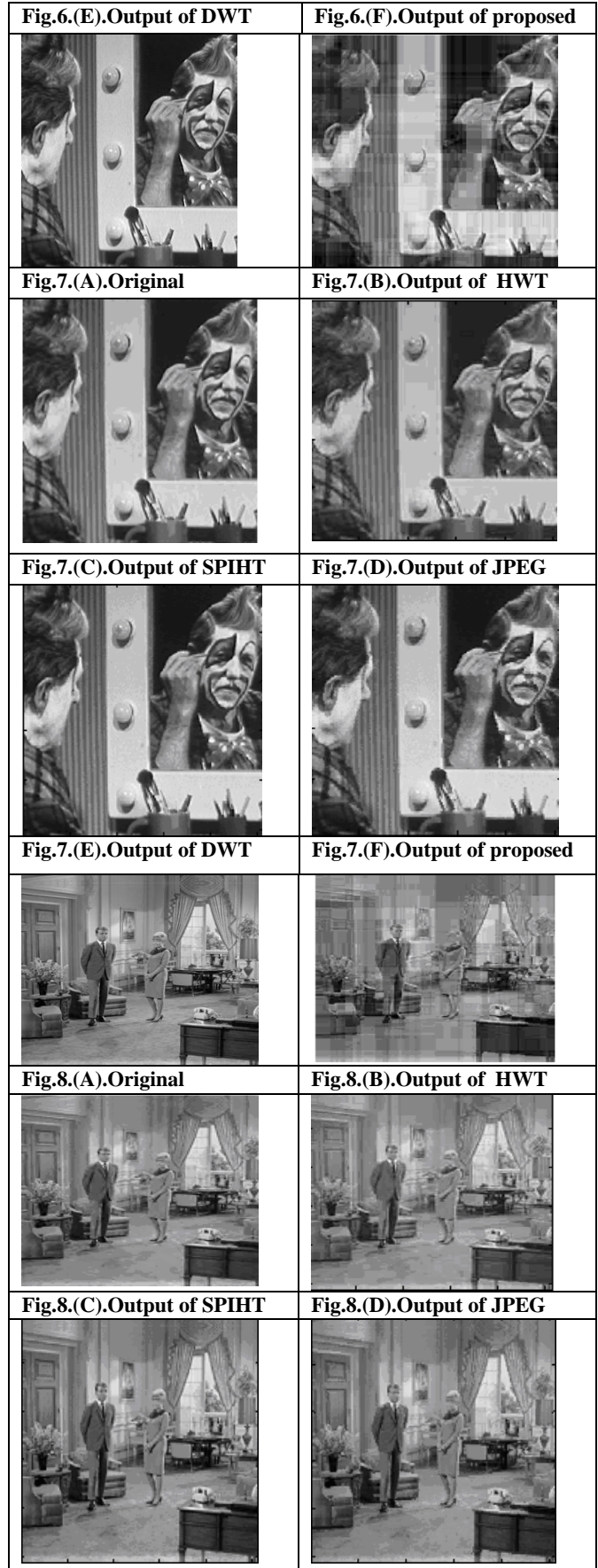
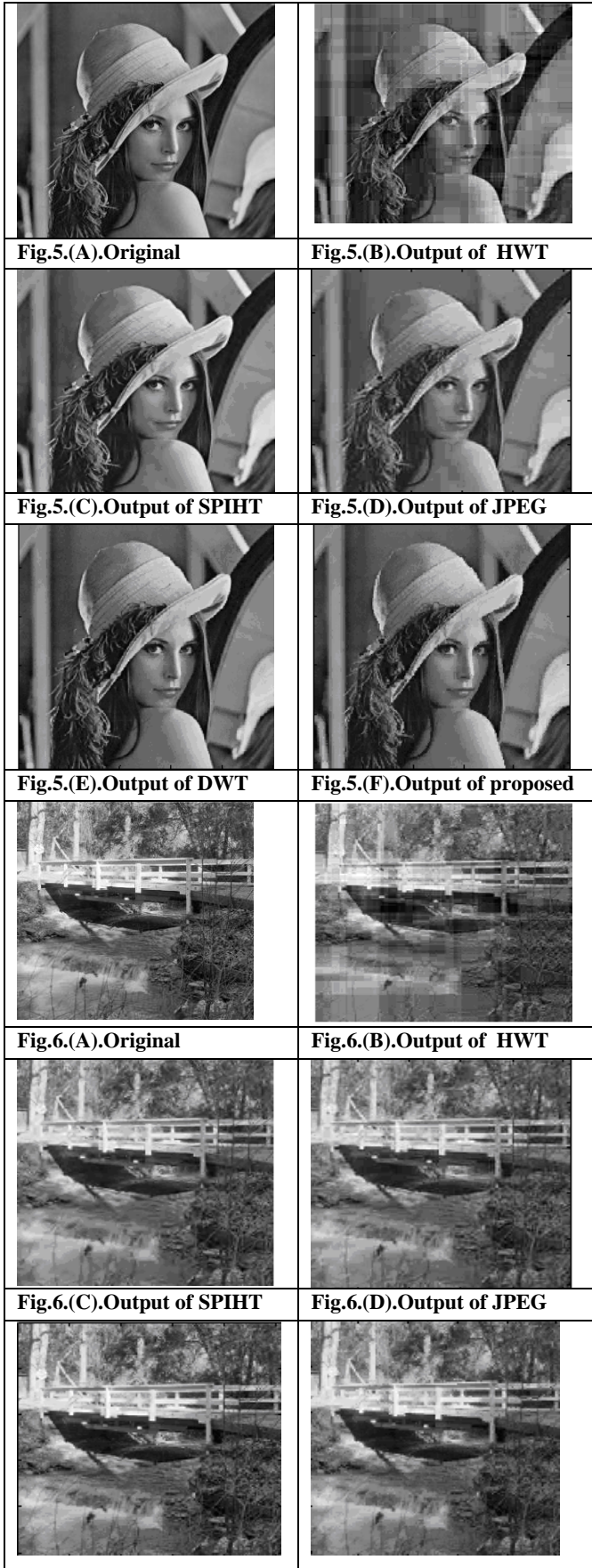


Fig.8.(E).Output of DWT	Fig.8.(F).Output of proposed
--------------------------------	-------------------------------------

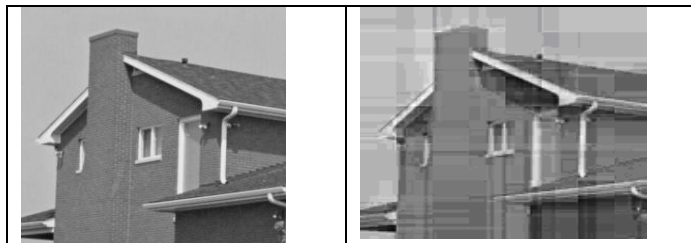


Fig.9.(A).Original

Fig.9.(B).Output of HWT



Fig.9.(C).Output of SPIHT

Fig.9.(D).Output of JPEG

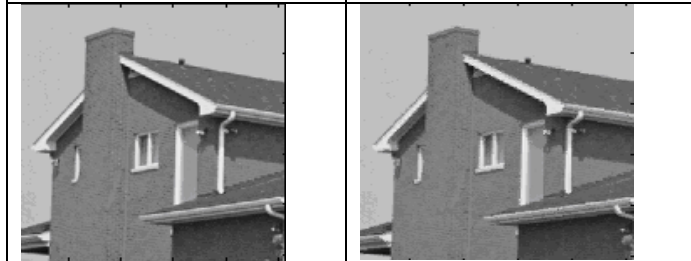


Fig.9.(E).Output of DWT

Fig.9.(F).Output of proposed

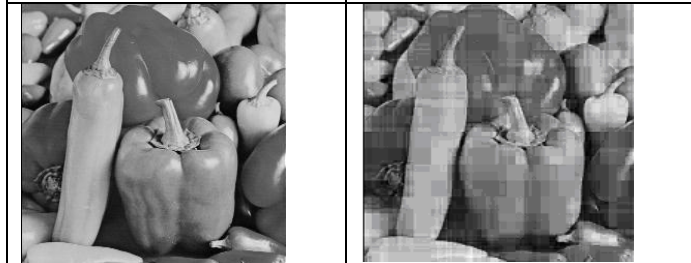


Fig.10.(A).Original

Fig.10.(B).Output of HWT



Fig.10.(C).Output of SPIHT

Fig.10.(D).Output of JPEG



Fig.10.(E).Output of DWT

Fig.10.(F).Output of proposed

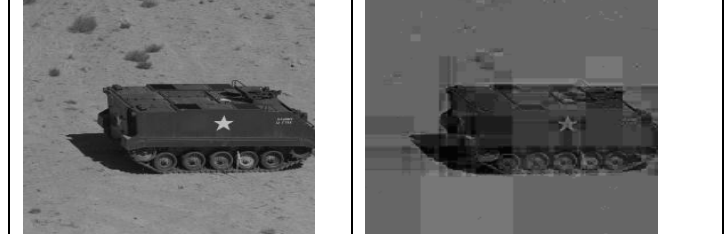


Fig.11.(A).Original

Fig.11.(B).Output of HWT

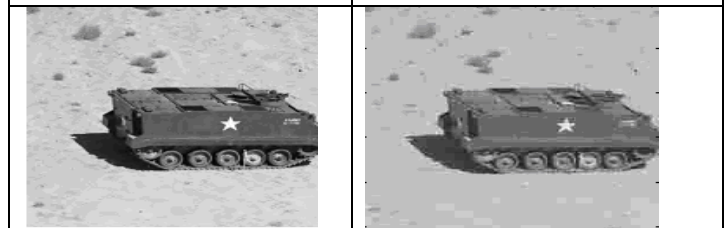


Fig.11.(C).Output of SPIHT

Fig.11.(D).Output of JPEG

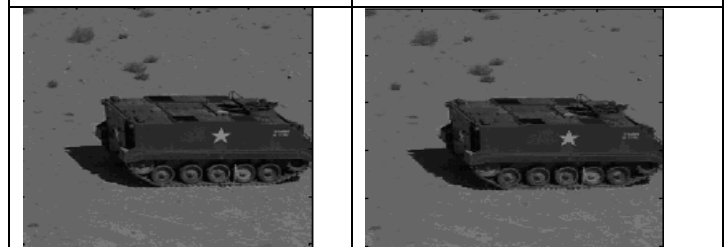


Fig.11.(E).Output of DWT

Fig.11.(F).Output of proposed

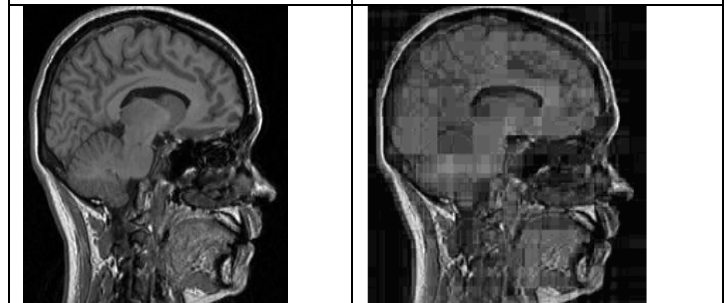


Fig.12.(A).Original

Fig.12.(B).Output of HWT

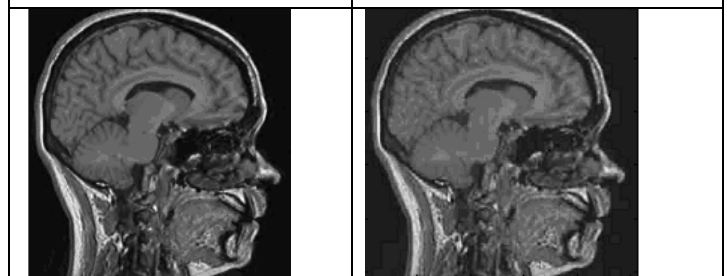


Fig.12.(C).Output of SPIHT	Fig.12.(D).Output of JPEG
Fig.12.(E).Output of DWT	Fig.12.(F).Output of proposed
Fig.13.(A).Original	Fig.13.(B).Output of HWT
Fig.13.(C).Output of SPIHT	Fig.13.(D).Output of JPEG
Fig.13.(E).Output of DWT	Fig.13.(F).Output of proposed
Fig.14.(A).Original	Fig.14.(B).Output of HWT

Fig.14.(C).Output of SPIHT	Fig.14.(D).Output of JPEG
Fig.14.(E).Output of DWT	Fig.14.(F).Output of proposed

4. Conclusion

This paper proposed a novel image compression algorithm based on block truncation coding (BTC) which provide higher compression ratio as compared to existing lossy compression techniques. This algorithm was compared with existing lossy image compression techniques such as HWT, JPEG, SPIHT, and DWT. This algorithm is simple and efficient. It can be used for high quality image compression in applications such as cartoons where picture quality is very poor. But definitely it is not suitable at all in applications such as medical imaging and other related low quality application. Simulation & experimental results demonstrated that proposed algorithm outperform as compared to existing lossy image compression techniques. In future this work can be extended to video compression.

Reference

- [1]. T. Acharya, A. K. Ray, "Image Processing: Principles and Applications", John Wiley & Sons, pp.351-368. 2005,
- [2]. R. C. Gonzolez, R. E. Woods, S. L. Eddins, "Digital Image Processing Using Matlab", Prentice Hall, 2004.
- [3]. Algorithms for Image Processing and Computer Vision, J. R. Parker, John Wiley & Sons, Inc., 1997.
- [4]. T.Sreenivasulu reddy, K.Ramani, S.Varadarajan and B.C.Jinaga, Image Compression Using Transform Coding Methods, IJCSNS International Journal of Computer Science and Network Security, VOL. 7 No. 7. July,2007
- [5]. Ahmed, N., Natarajan, T., and Rao, K.R. **Discrete Cosine Transform**, *IEEE Trans. Computers*, vol. C-23, Jan. pp. 90-93,1974,
- [6]. Ken Cabeen and Peter Gent, Image Compression and the Cosine Transform, Math 45, College of the Redwoods,<http://online.redwoods.cc.ca.us/instruct/darnold/laproj/fall98/pken/dct.pdf>

- [7]. Subhasis Saha, "Image Compression - from DCT to Wavelets : A Review", www.acm.org/crossroads/xrds6-3/sahaimgcoding.htm
- [8]. Peggy Morton HP Authorized, Image Compression Using the Haar Wavelet transform, <http://online.redwoods.cc.ca.us/instruct/darnold/laproj/Fall97/PMorton/imageComp3/>
- [9]. G. K. Wallace, 'The JPEG Still Picture Compression Standard', Communications of the ACM, Vol. 34, Issue 4, pp.30-44, 1991.
- [10]. C. Cuturicu, 'A note about the JPEG decoding algorithm', <http://www.opennet.ru/docs/formats/jpeg.txt>, 1999.
- [11]. Said, A. and Pearlman, W. A. A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, *IEEE Trans. CSVT*, vol. 6, no. 3, pp.243-250, June 1996, http://ipl.rpi.edu/pub/EW_Code/SPIHT.ps.gz
- [12]. B. E. Usevitch, 'A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000', *IEEE Signal Processing Magazine*, vol. 18, pp. 22-35, Sept. 2001.
- [13]. Kotsas P, Piraino DW, Recht MP, Richmond BJ: "Comparison of adaptive wavelet-based and discrete cosine transform algorithms in image compression," *Radiology pp.*193-331, 1994.
- [14]. Lawson S. and Zhu J Image Compression Using Wavelets and JPEG2000", *Comm, Electronics and Commn. Engg. Journal*, June, 2002.
- [15]. Lewis, A. S. and Knowles, G. Image Compression Using the 2-D Wavelet Transform, *IEEE Trans. IP*, vol. 1, no. 2, pp. 244-250, April 1992.
- [16]. Seungjong Kim and Jechang Jeong, Image Compression Using the Wavelet Transform and Context-Based Arithmetic Coding, *Proc. SPIE: Second International Conference on Image and Graphics Vol. 4875*, pp. 133-140, 2002.
- [17]. Image Data Compression: Block Truncation Coding, Los Alamitos, CA: IEEE Computer Society Press, 1995.
- [18]. Image Compression Using Block Truncation Coding, Delp, E.J., and Mitchell, O.R., *IEEE Transaction on Communication*, Vol.27, No. 9, pp.1335-1342, Sept. 1979.



Shyam Lal received B.Tech. (with Hons.) in Electronics & Communication Engineering from Bundelkhand Institute of Engineering & Technology (B.I.E.T) Jhansi (U.P.)-India and M.Tech.(with Hons.) in Electronics & Communication Engineering from National Institute of Technology, Kurukshetra (Haryana)-India in year 2001 & 2007, respectively. He is pursuing Ph.D. degree in

the area of Digital Image Processing from Department of Electronics & Communication Engineering, Birla Institute of Technology, Mesra, Ranchi (Jharkhand)-India

He is Associate Professor in the department of Electronics & Communication Engineering, Moradabad Institute of Technology, Moradabad (U.P.)-India. He has published more than 33 papers in the area of Digital Image Processing and Wireless Communication & Computing at National/International Level. He is life member of Indian Society of Technical Education (ISTE), New Delhi-India (LM-39989), International Association of Engineers (IAENG), Hong Kong (M.No.-103480) and Life member International Association of Computer Science and Information Technology (IACSIT), Singapore (LM- 80333445). He has more than eight years teaching experience. His area of interest includes Digital Image Processing, Digital Signal Processing and Wireless Communication.



Mahesh Chandra received B.Sc. from Agra University, Agra (U.P.)-India in 1990 and A.M.I.E. from I.E.I., Kolkata (W.B.)-India in winter 1994. He received M.Tech. from J.N.T.U., Hyderabad-India in 2000 and Ph.D. from AMU, Aligarh (U.P.)-India in 2008. He has worked as Reader & HOD in the Department of Electronics & Communication Engineering at S.R.M.S. College of Engineering and Technology, Bareilly (U.P.)-India from Jan 2000 to June 2005. Since July 2005, he is working as Reader in the Department of Electronics & Communication Engineering, B.I.T., Mesra, Ranchi (Jharkhand)-India. He is a Life Member of ISTE, New Delhi-India and Member of IEI Kolkata (W.B.)-India. He has published more than 24 research papers in the area of Speech, Signal and Image Processing at National/International level. He is currently guiding 04 Ph.D. students in the area of Speech, Signal and Image Processing. His areas of interest are Speech, Signal and Image Processing.



Gopal Krishna Upadhyay received B.Sc. (Maths) from Kanpur University, Kanpur (U.P.)-India in 1992., M.Sc. (Electronics) from V.B.S. Purvanchal university Jaunpur (U.P.)-India in 1994 and Ph.D.(Solid State Physics) from V.B.S. Purvanchal University Jaunpur (U.P.) in 2001. He worked as Lecturer in the Department of Physics, K.A.P.G.A. Allahabad (U.P.)-India from July 1994 to 2003 and Asstt. Professor in Department of Physics of United College of Engg., Allahabad (U.P.)-India from July 2000 to July 2003. He had also worked as Professor & Dy. Director in T.M.I.M.T, Teerthankar Mahaveer University, Moradabad (U.P.)-India. He is currently working as Director, SSITM Kasganj-Moradabad (U.P.)-India. He has more than 15 years of teaching experience. He has published more than 30 papers in the National/International level. He has also published 4 books. He is currently guiding 05 Ph.D. students. His area of interest is solid state devices, computing & image processing.

Expert System For Online Diagnosis of Red-Eye Diseases

Muhammad Zubair Asghar(1), Muhammad Junaid Asghar(2)

Abstract-- This paper describes Expert System (ES) for online diagnosis and prescription of red-eye diseases. The types of eye diseases that can be diagnosed with this system are called Red-eye diseases i.e. disease in which red-eye is the common symptom. It is rule based web-supported expert system, assisting ophthalmologists, medical students doing specialization in ophthalmology, researchers as well as eye patients having computer know-how. System was designed and programmed with Java Technology. The expert rules were developed on the symptoms of each type of Red-eye disease, and they were presented using tree-graph and inferred using forward-chaining with depth-first search method. User interaction with system is enhanced with attractive and easy to use user interfaces. The web based expert system described in this paper can detect and give early diagnosis of twenty Red-eye diseases. This WES can be extended to diagnose all types of eye-diseases.

Keywords: Expert System, Red Eye, Ophthalmologist, Diagnose, Artificial Intelligence, Knowledge, Database.

1. Introduction

This is web-based expert system called WES (Web based Expert System) for Diagnoses of Red Eye, to diagnose eye diseases infecting Pakistani population having red eye as common symptom. WES is an enhanced version of the CADRE expert system [6] ,It contains a revised and extended knowledge base as well as more up-to-date inference mechanism.

The present work describes following improvements over the other similar expert systems in the field.

- i. All previous Red Eye ES are non-internet and non-GUI based systems. The present work is development of web based system having easy to use GUI for the user interaction.
- ii. Some of the earlier expert systems have grown up only at the prototype stage. There was obvious potential for more practical nature Red Eye based ES in ophthalmology and WES has been developed to achieve that.
- iii. Previous systems were able to deal with fewer Red Eye diseases, resultantly backward chaining of rules was used. No. of Red Eye diseases that our system can diagnose (25) are more than earlier systems, so we did use forward chaining with depth first search method.
- iv. The most important characteristic of this system is that it doesn't need to have the answers to every input question in order to reach a conclusion. The system will

not ask the same question twice, For example, there are multiple red eye diseases that contain one/more common symptoms, so the responses given earlier will automatically be applied to new diagnoses.

An expert system is an Artificial Intelligence based computer program, which acts like a human expert in particular area of knowledge. It has three main components i.e. a knowledge base (KB), an inference engine (IE) and control strategy (CS) [2]. Knowledge structured in the form of IF-THEN rules is stored in KB. This knowledge is processed by inference engine under supervision of control strategy for achieving expert advice. A Web Based Expert System(WES) is a collection of computer software and hardware components that have been properly selected, designed, developed, combined and configured in order to deliver a service that emulates in an effective and reliable manner the reasoning process of domain experts over the Web[3].

1. Related Work

In medicine, *red eye* is a non-specific term to describe an eye that appears red due to illness, injury, or some other condition. "Conjunctival injection" and "bloodshot eyes" are two forms of red eye. Since it is a common affliction, it is unsurprising that primary care doctors often deal with patients with red eyes in their practices. The goal of the primary care doctor when presented with a red eye is to assess whether it is an emergency in need of referral and immediate action, or instead a benign condition that can be managed easily and effectively. Red eye usually refers to hyperemia of the superficial blood vessels of the conjunctiva, sclera or episclera, and may be caused by diseases or disorders of these structures or adjacent structures that may affect them directly or indirectly [1].

2.1 Causes

There are many causes of a red eye including conjunctivitis, blepharitis, acute glaucoma, injury, subconjunctival hemorrhage, inflamed pterygium, inflamed pinguecula, and dry eye syndrome[1].

2.2 Investigation

Some signs and symptoms of red eye represent warnings that the underlying cause is serious and requires immediate attention. The person conducting a thorough eye examination should be attentive to the warning signs and symptoms during the eye exam. There are six danger signs: conjunctival injection, ciliary flush (circumcorneal injection), corneal edema or opacities, corneal staining, abnormal pupil size, and abnormal intraocular pressure[1].

There are approximately one hundred and fourteen (114) possible causes/medical conditions of Red Eye[1]. These causes/symptoms are used as rules in our web based medical expert system for diagnoses of Red Eye.

Due to rising cost of health care, web based medical expert system has proved useful for assisting medical practitioners and helping patients to self manage Red Eye. A few medical expert systems have been reported earlier for Red Eye Diagnosis. Marfina ulduna[4]. Ibrahim, F; Ali, J.B developed expert system for early diagnoses of eye diseases (including some cases of Red Eye) infecting the Malaysian population [5]. CARDRE ES was developed by Zubair et. al.[6] for diagnoses of Red eye diseases (20-diseases) infecting Pakistani population.

ES can be developed by two ways: i) Shell based approach ii) using some language like prolog, lisp, Java etc.: The requirements of the specific application determine the value of the capabilities and features required. More than 160 ES shells are available commercially for ES development. EXSYS (Schmoltdt & Martin, 1989) and JESS (java expert system shell [8], which is java version of CLIPS ES shell, are mostly used for web based expert system development. The JESS source code can be embedded into java programs. JESS provides KB development environment and inference capability. However additional capabilities for GUI, images as well as JAVA Database Connectivity (JDBC) have to be programmed. The same is true for most of other ES shells like CLIPS and EXSYS.

3. Objectives of Current Research

The need for web based decision support and expert systems has been felt world wide as they are capable to provide comprehensive and up-to-date information and consultation in interactive and user friendly manner. Web based system has been developed to fulfill the following objectives:

- To develop an online ES that may provide free consultation about Red eye diseases.
- To assist ophthalmologists for diagnosing various diseases associated with red eye.
- All health care professionals including, ophthalmologists, medical students, pharmacists can keep their knowledge up-to-date regarding “Red-eye

diagnoses and treatment”, as its knowledge base external database is updated on regular basis.

4. Methodology:

Phases/steps carried out in developing WES do include: 1.Problem definition/Scope identification. 2. Knowledge acquisition. 3. Knowledge representation. 4. Coding. 5. Testing and implementation.

4.1. Problem Definition/Scope Identification.

An expert system needs precise domain. The domain must be well organized and well understood. In diagnosis domain, as number of disorders (diseases) increase linearly, the number of possible diagnoses increase exponentially (i.e. there are more combinations of diagnoses to consider). This type of growth in the total no. of solutions is called combinatorial explosion[10]. Clearly, Red Eye diagnoses are an appropriate domain for expert system development. This WES has a narrow domain of Red eye diseases. Following list shows some sample red eye diseases which can be diagnosed by WESDRE.

- 1) Blepharitis, (2) Bacterial keratitis,(3) Endophthalmitis, (4) Episcleritis, (5) Scleritis, (6) Chalazion, (7) Corneal ulcers, (8) Uveitis, (9) Ocular Rosacea, (10) Ectropion, (11) Entropion, (12) Foreign body and Red eye, (13) Viral Conjunctivitis, (14) Orbital Cellulitis, (15) Allergic Conjunctivitis, (16) Iritis, (17) Acute Angle-closure Glaucoma, (18) Bacterial Conjunctivitis, (19) Herpes Zoster, (20) Dry Eye Syndrome, (21) Episcleritis, (22) Vogt-Koyanagi-Harada syndrome (23) Choroidal melanoma.

4.2. Knowledge Acquisition.

Knowledge Acquisition includes three activities: choosing what knowledge is needed; obtaining and interpreting the knowledge; and modeling the knowledge. Some commonly used approaches are direct interviews, observations Ishikawa diagrams, case studies [7]. Direct interviews and observations were used for KA. The KA for this system consisted of several interviews with ophthalmologists, making observations and getting historical data from various ophthalmology clinics, depts. and wards in DHQ teaching hospital D.I.Khan, free eye camps and ophthalmology labs of medical college. Knowledge acquisition process lasted for four months.

4.3. Knowledge Representation

There are numerous approaches for knowledge representation depending on nature of problem domain. As this is a rule based system, so IF-THEN style rules are used because they are easy to understand and enhance. The rule has two components: IF<situation THEN<suggestion> The

IF-part (antecedent/left hand side) suggests describes a situation under which the rule is applicable. The THEN part (consequent/right hand side) suggests an action to be performed under the situation (for action rules) or a plausible inference to make under the situation(for inference rules)[7].

For example:

Sample JESS rule taken form WESDRE is given below

```
Jess>(defrule red-eye-rule-1
(symptom (name ? blapheritis) (upp_lid_pain true) (eye_red true)
(sandy_feeling true)(fever yes) =>(max cahnces of blaphiritis, with CF ?too_high). This particular If-rule shows nested and conditions for red-eye disease called blepharitis.
```

For present application, appropriate option is JESS, Java Servlets/JSP based , JDBC-ODBC bridge for database access, image composing tools like image composer, Segate Crystal reports as reporting tool. WESDRE Web based model is shown in Fig.1

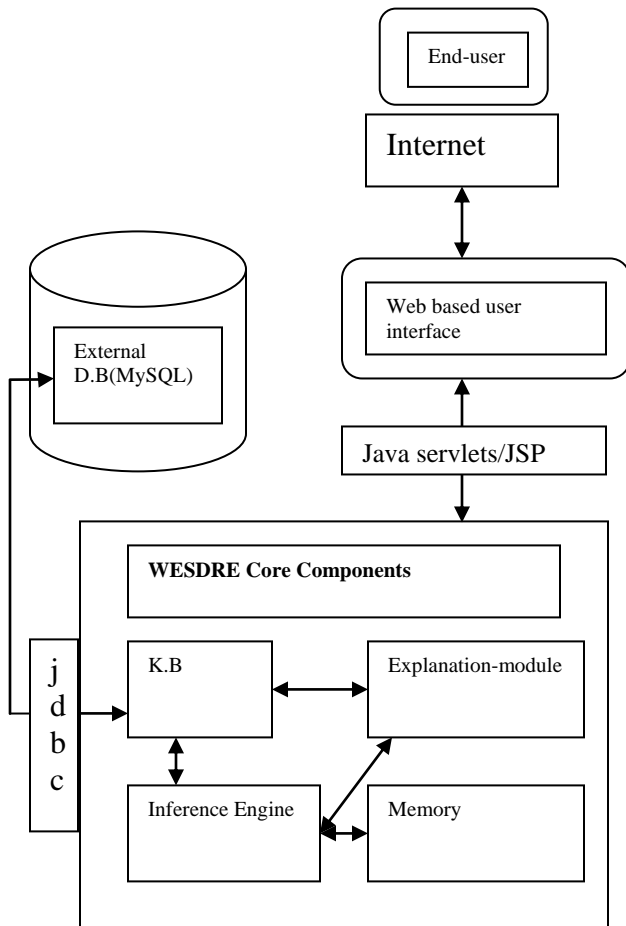


Fig.1 WES Model

4.4. Testing and implementation

System is tested to ensure that it provides good quality decisions. Reasoning technique is tested for correctness i.e. whether it attains the desired accuracy rate. During testing omitted rules, incorrect rules, overlooked special cases are checked. Association of rules and missing relationships are also checked. Thorough testing verifies that all paths in the knowledge base are covered. ES was tested by two Eye Specialists/ophthalmologists. The prototyping approach was used to implement WES. A small domain of the knowledge acquired was implemented in JESS and presented to the ophthalmologists. They tested the prototype using different scenarios. They recommended additions, deletions or changes from the conclusions given by WES. Once the Ophthalmologists agreed with the recommendations given by the ES, new prototype with more knowledge was developed and presented again to experts for testing. The procedure continued until all the acquired knowledge was included in WES knowledge base. WES is planned to be launched on the web using Apache server.

5. WES: How Does It Work?

The WES knowledge base includes over 300 facts and 400 rules for diagnosing all types Red Eye diseases. Proposed system strictly incorporates the diagnostic criteria followed by human experts. There are twenty five diseases associated with “Red Eye” with each disease having average of 15 to 20 symptoms. System is able to diagnose all twenty five diseases of Red eye. WES working model is comprised of following modules: symptom analysis phase-I, symptom analysis phase-II, disease selection with appropriate percentage, medicine selection for disease diagnosed, knowledge base, user interface design.

5.1 Phase-I

When system is turned on and option “consultation” is selected from the main menu then all consultation begins in question answer format. User answers “yes” or “no” when “yes” is clicked/checked then risk factor retains its previous value. e.g. Do you feel that your eyelashes are turning inwards?

If user checks “yes”, then following action takes place

Assign bleph_fact:=bleph_fact+10

If “No” or “Unknown” is checked then bleph_fact retains its previous value. In this Question answer session if certainty factor of one/more diseases gets increased from 40 then their follow-up question are asked. This is beginning of phase-II[6].

5.2 Phase-II

In this phase, detailed/remaining/follow-up questions of only those diseases are asked whose certainty factors are greater than 40. Thus no. of questions (symptoms) in this phase are less than phase-I. User again answers yes/no/unknown to follow up questions. At the end of this follow up session disease(s) is/are diagnosed in the form of percentage i.e. possible disease(s) is/are listed along with percentage(s) that a patient can suffer from[6]. One of the Graphical diagnostic reports is shown in Fig. 3.

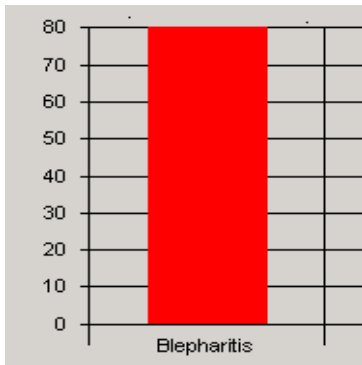


Fig. 3 Graphical Diagnostic Report of WES.

5.3 Knowledge Base Design

A web based knowledge base is designed that assists knowledge engineer for representing facts and rules. JESS knowledge acquisition and representation editor (DRAMA 2.0) with object oriented graphic tools makes it very easy. Facts and rules regarding disease symptoms, CF values, treatment, prevention methods/actions can entered online by the experts who are authorized to access it. Disease Tree diagram taken from WESDRE KB is shown in Fig: 2



Fig. 2. Disease Tree Diagram of WES KB.

It shows that WES's scope

5.4 External Data Base

External Database is created in MySql for storage and retrieval of symptoms, diseases, treatment, prevention information, and rules associated with WES knowledge base.

5.5 User Interface

The acceptability of an expert system depends to a great extent on the quality of user interface. The Web based user interface is supported by java servlets, where user can respond to questions asked by the system. Java servlet technology provides a simple, consistent mechanism for extending the functionality of web server. A servlet can almost be thought of as an applet that runs on the server side. Servlets provide a component based platform-independent method for building Web-based applications, without the performance limitations of CGI programs [9].

6. Results and Discussion

When the Doctor/patient visits WES, series of questions appear on computer screen. This is phase-I of "symptoms analysis". After this system asks follow-up questions, this is phase_II of symptom analysis. The end result of this phase is a "diagnostic report" about the patient under examination, suggesting a possible disease. It also suggests proper management and treatment for that particular disease. Learned Patients can get benefits from this system without assistance of Ophthalmologist. Table-1 shows a sample session with WES[6]. This ES is capable of giving answers to questions like "EXPLAIN" and "WHY" at each and every step during interaction with system. After detailed dialogue system comes with following suggestion. You have chances of suffering from a red-eye disease "Blepharitis" Certainty factor (CF)= 80%.

7. Conclusion

WES is an extension its previous version CADRE, which was developed by ICIT, GU, D.I.KHAN students under the supervision of Zubair Asghar[6]. It benefits common man (eye patient in this case) through the use internet based ES consultation. It also assists all ophthalmologists/medical students for diagnosing Red eye diseases. Adding new facts and rules to WES is very easy for other diseases of eye are very easy through JESS knowledge acquisition editor.

Table-1: Sample session with WES

WES	Do you feel eye-pain, especially with eye movement?
Patient	Yes.
WES:	Is there any painful swelling of the upper eyelid?
Patient	Yes.
WES:	Eyes red or not?
Patient	Yes.
WES:	Is there any irritation, gritty or sandy feeling?
Patient	No.
WES:	Do you feel fever, malaise or headache?
Patient	Yes
WES:	Is there any light sensitivity?
Patient	No
WES:	The eye movement is restricted or not?
Patient	Yes.
WES:	Eye lids appear shiny and appear red or purple in color?
Patient	Yes.
WES:	Do you feel partial loss of vision?
Patient	Yes.

7. Mario A Garcia, Amit J.Gandhi, Tinu ingh, Texas A&M university. ESDIABETES(An Expert system in diabetes,JCSC 16,3, March 2001).
8. <http://herzberg.ca.sandia.gov/jess/>
9. M.Watson, Intelligent Java Applications, Morgan Kaufmann Publisher, anFrancisco,1997,pp(115-125).
10. W.D Potter, X.Deng, J.Li, M.Xu,Y.Weii,Lappas, "A web based Expert System for Gypsy Moth Risk Assesment", Artificial Intelligence Center, GSRC 111, University of Georgia, Athens, GA, 30605.

⁽¹⁾**Muhammad Zubair Asghar,**

*Institute Of Computing and Information
Technology, Gomal University, D.I.Khan.*

Zubair_icit@yahoo.com

⁽²⁾**Muhammad Junaid Asghar,**

*Department of Basic Medical
Sciences ,Gomal University D.I.Khan.*

Email:mrjunay@hotmail.com

8. Future Work

In future this system will be extended to diagnose all Eye diseases. WES will be made ready for next stage, where national/regional languages like urdu can be used for interaction with it on the web. This will make available to the patients without language barrier.

Reference

1. <http://www.wrondiagnosis.com>.
2. Harvindar, Kamal, Sharma, "A web based fuzzy expert system for integrated PEST management in soybean" International journal of information technology, 2002.
3. Atkins, C., Bunse, C., &Gro, H.-G"A Web-based component oriented design",2002).
4. "An Expert System for Red Eye Diseases"
http://eric.ed.gov/ERICWebPortal/Home.portal?_nfpb=true&_pageLabel=RecordDetails&ERICExtSearch_SearchValue_0=ED415753&ERICExtSearch_SearchType_0=eric_accno&objectId=0900000b80130a42
5. Ibrahim, F; Ali, J.B(2001)"An expert system for early diagnoses of eye diseases (including some cases of Red Eye) infecting the Malaysian population,IEEE,2001.
6. "Dr. A. Rashid, Zubair Asghar, Tania Karim, "CADRE", master thesis 2003 , ICIT,G.U.

Handwritten Devnagari Numeral Recognition using SVM & ANN

Sandhya Arora¹, Debotosh Bhattacharjee², Mita Nasipuri², M. Kundu² and D. K. Basu³ and L.Malik⁴

¹Dept. of CSE & IT, Meghnad Saha Institute of Technology
kolkata, 700150,India

¹sandhyabhagat@yahoo.com

²Department of Computer Science and Engineering, Jadavpur University
Kolkata, 700032,India

³AICTE Emeritus Fellow, Department of Computer Science and Engineering, Jadavpur University
Kolkata, 700032,India

⁴Dept. of Computer science, G.H. Rasoni college of Engineering
Nagpur, India

Abstract: This paper proposes a system for recognizing offline Handwritten Devnagari numerals using support vector machine and artificial neural networks. The proposed system classifies numeral, in two stages. Various preprocessing operations are performed on the digitized image to enhance the quality of the image. It involves image acquisition and numeral image extraction, binarization, scaling, thinning, smoothing and noise removal. Feature extraction where some statistical and structural features, such as - shadow based features, zone based directional features, zone based centroid features and view based features, are extracted after preprocessing. Finally, the classification phase takes place in two stages. In first stage, numerals are classified using MLP. Unrecognized numerals of first stage, are then classified in second stage by SVM using one-against-all technique to classify 10 handwritten devnagari numeral shapes. The proposed system has been tested on 18300 data samples. The system has achieved nearly 93.15% recognition rate.

1. Introduction

In Optical Character Recognition [OCR], a character/numeral which has to recognize can be machine printed or handwritten. There is extensive work in the field of handwriting recognition, and a number of reviews exist. Handwritten numeral recognition is an exigent task due to the restricted shape variant, unusual script style & different kind of noise that breaks the strokes in number or changes their topology. Recognize of is gaining wider importance today and is one of the benchmark problem in document analysis. As handwriting varies when person write a same character twice, one can expect enormous dissimilarity among people. These are the reason that made researchers to find techniques that will improve the knack of computers to characterize and recognize handwritten numerals.

Recognizing handwritten numerals is an important area of research because of its various application potentials. Automating bank cheque processing, postal mail sorting, job application form sorting, automatic scoring of tests containing multiple choice questions and other applications where numeral recognition is necessary. Some research has been done on the recognition of Roman, Arabic and Chinese numerals which is excellently reviewed in [1]. Le Cun et al [2] have developed an algorithm for identifying Arabic numerals with a high recognition rate. Few works is available for Devnagari numeral recognition using Neural Networks but none for SVM. The first research report on handwritten Devnagari characters and numerals was published in 1977 [3] but not much research work has been done after that. Hanmandlu and Murthy [4] proposed a Fuzzy model based recognition of handwritten Hindi numerals and they obtained 92.67% accuracy. Bajaj et al [5] employed three different kinds of features namely, density features, moment features and descriptive component features for classification of Devnagari Numerals. They proposed multi-classifier connectionist architecture for increasing the recognition reliability and they obtained 89.6% accuracy. Bhattacharaya et al. [6] proposed a Multi-Layer Perceptron (MLP) neural network based classification approach for the recognition of Devnagari handwritten numerals and obtained 91.28% recognition accuracy. An excellent survey of the area is given in [7].

In this paper, a system for off-line recognition of handwritten Devnagari numerals is proposed using ANN and SVM classifiers. There have been several attempts for OCR of Indian printed characters but very few of these are for recognition of handwritten numerals, and none using SVM. We applied ANN and SVM at different stages of classification. In first stage, the numeral image

is preprocessed (section 3) and four features namely: shadow based features, zone based directional features, zone based centroid features and view based features are extracted (section 4). These features are then fed to MLP's (section 5), designed separately for all four features for recognition. Results of four MLP's are combined using weighted majority scheme. Numerals not classified by MLP, in first stage are classified using SVM (section 5), in second stage. The section 6 provides discussion regarding results and conclusion is summarized in section 7.

2. Challenges in Handwritten Devnagari Numeral Recognition

Devnagari is the most popular script in India. Hindi is written in Devnagari script. Nepali, Sanskrit and Marathi are also written in Devnagari script. Further, Hindi is the national language of India and Hindi is the third most popular language in the world. According to a recent survey English is being used by 125.3 million people, Bengali is used by 91.1 million people, Telugu and Marathi is being used by 85 million and 84.2 million people in India. Thus, work on recognition of handwritten Devnagari numerals is of considerable practical importance. Because of the writing styles of different individuals, numerals can have different shapes. Handwritten Devnagari numeral recognition is an exigent task due to the restricted shape variant, unusual script style & different kind of noise that breaks the strokes. As a result recognition of handwritten numerals becomes a difficult task. Some sample handwritten Devnagari numerals are shown in Figure 1 to give some idea of the vast disparity in writing styles for different characters. Figure 2 shows some of the handwritten samples (phone numbers) written by four different writers.

Numerals	Handwritten Devnagari Numerals				
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					

Figure 1. Examples of Handwritten Devnagari Numerals

९१-९८२२७-०९०७२ ९९-९४२२९-८९४०२
 ०७९८४-२७६४०९ ९९-९४२२२-९०३४४

Figure 2. Numeral Samples

2. Process overview

The character recognition system is usually validated by running them on independent test sets, on which the systems have not been trained. For these tests to be conclusive, the set should include a fairly large number of samples to reflect the variety of writing styles that are found in real-life applications. The task of the recognition of handwritten numerals has been broken down into the following steps:-

- (i) binarization of sample image;
- (ii) thinning of the image;
- (iii) smoothing;
- (iv) normalization of the image to a standard size;
- (v) feature extraction;
- (vi) recognition.

To enable recognition, steps (i)–(iv) are applied on a training set of all 10 numerals as part of the pre-processing. While performing feature extraction, simultaneously the Knowledge Base of reference features is created, discussed in section 4.

3. Preprocessing

The preprocessing steps remove any noise, distortions in the input character and convert the character in a form processed and recognizable by the system. The

preprocessing steps are performed is for improving the quality of images for ensuring better quality in the subsequent processing of image. In the scanning process, some distortion in images may be introduced due to pen quality, light hand handwriting, poor quality of the paper on which the numerals are written etc. The preprocessing steps performed in this work are consists of the following:-

3.1. Binarization

Frequently, binarization is carried out before the character recognition phase. Ideally an input character should have two tones, i.e., black and white pixels (commonly represented by 1 and 0, respectively). Image binarization converts an image of up to 256 gray levels into a two-tone image. The goal of this step is to identify a threshold value dynamically that would help to distinguish between image pixels that belong to text and those that belong to the background. The threshold value identified would be completely dependent on the nature and the properties of the documents, the contrast level, the difference of contrast between the foreground and background pixels and their concentration in the document. The methodology would be applied to gray-level image (range of pixel values 0 to 255).

1. Take the threshold to be 128 (midway between 0 and 255).
2. Take all the pixels with grayscale values above 128 as background and all those with values below 128 as foreground.
3. Find the mean grayscale value of background pixels as bMean and that of all foreground pixels as fMean.
4. Find the average of bMean and fMean and make this the new threshold.
5. Go back to step 2 and continue this process of refining the threshold till the change of the threshold from one iteration to next becomes less than 2% of the range of 0 to 255.

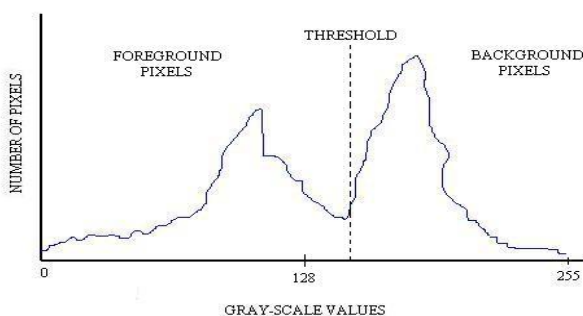


Figure 3. Finding the threshold

3.2. Thinning and smoothing

A two-tone digitized image is defined by a matrix A, whose element a_{ij} is either 1 if character is present or 0 otherwise. Iterative transformations are applied on A to obtain the thinned character. We have used a thinning algorithm of Ref. [8,17], by which a point at a binary pattern consists of successive deletion of dark points (i.e. changing them to white points) along the edges of the pattern until it is thinned to a line. This results some redundant pixels. To remove this redundancy we have applied certain masks [17], which gives one pixel-wide skeleton.

3.3 Size Normalization

Normalization is thus the process of equating the size of all numerals in order to extract features on the same footing. To achieve this, we use standard Affine Transformation to perform a linear mapping from 2D coordinates to other 2D coordinates that preserves the “straightness” and “parallelness” of lines. Affine transformation can be constructed using sequences of translations, scales, flips, rotations and shears. Image is scaled in 100x100 pixel resolution.

4. Feature Extraction

In the following, we discuss the extracted features for classifiers. These features are used in MLP and SVM classifiers discussed in section 5.

4.1 Shadow Features

Shadow is basically the length of the projection on the sides. For computing shadow features [10, 11] on scaled binary image, the rectangular boundary enclosing the numeral image is divided into eight octants. For each octant shadows or projections of numeral image segment on three sides of the octant dividing triangles are computed so, a total of 24 shadow features are obtained. Each of these features is divided by the length of the corresponding side of the triangle to get a normalized value.

4.2 Zone based directional features of Character Contour

Chain code provides the points in relative position to one another, independent of the coordinate system. In this methodology, we first find the contour points of the scaled image, and then direction chain coding of connecting neighboring contour pixels, and the outline coding are captured [13]. If c_f denote a contour chain using Freeman codes d_k , such that, $C_f = d_1 d_2 d_3 \dots d_k \dots d_n$

Where $dk \in \{0,1,2,3,4,5,6,7\}$, and n is the length of chain.

We divide the contour image in 5×5 blocks. In each of these blocks, the frequency of the direction code is computed and a histogram of chain code is prepared for each block. Thus for 5×5 blocks we get $5 \times 5 \times 8 = 200$ features for recognition.

4.3 View based features

This method is based on the fact, that for correct character-recognition a human usually needs only partial information about it – its shape and contour. This feature extraction method [11], examines four “views” of character. The view is a set of points that plot one of four projections of the object (top, bottom, left and right). In the considered examples, eleven uniformly distributed characteristic points are taken for each view. These quantities are normalized so that their values are in the range $\langle 0, 1 \rangle$. Now, from 44 obtained values the feature vector is created to describe the given numeral, and which is the base for further analysis and classification.

4.4 Zone based Centroid Features

For extracting the feature, the zone-based hybrid approach is proposed. The most important aspect of the handwriting recognition scheme is the selection of a good feature set, which is reasonably invariant with respect to shape variations caused by various writing styles. The major advantage of this approach stems from its robustness to small variations ease of implementation. The zone-based feature extraction method gives good results even when certain preprocessing steps like filtering; smoothing and slant removing are not considered.

Definition 1. Centroid Feature Vector: The numeral image centroid is computed and the numeral image (100x100) is divided into 25 equal zones (20x20). Zone centroid is computed. This procedure is sequentially repeated for the entire zone present in the numeral image (50 features). There could be some zones having empty foreground pixels. Hence feature value of such zone in the feature vector is zero. Distance of zone centroids with image centroid is also calculated.

By the definition of centroid, the centroid of the image can be calculated as follows:

$$\begin{cases} x_c = \frac{\sum_{(x,y) \in p} xI(x,y)}{\sum_{(x,y) \in p} I(x,y)} \\ y_c = \frac{\sum_{(x,y) \in p} yI(x,y)}{\sum_{(x,y) \in p} I(x,y)} \end{cases}$$

where x_c, y_c are called as the coordinates of X-axis and Y-axis of numeral image p . A feature $\theta = (\theta_1, \theta_2, \dots, \theta_{100})$ is formed.

Input: Scaled and Thinned, Binarized Handwritten Numeral Image

Output: Extracted Features for Classification

Method Begins

Step1: Calculate the numeral image centroid x_c, y_c

Step2: Divide the image into 25 equal zones.

Step3: For each zone calculate the zone centroid x_i, y_i where $i=1,2,\dots,25$ (50 features)

Step4: Calculate the distance between zone centroid and image centroid as $x_c - x_i$ and $y_c - y_i$ (50 features)

Step5: Finally 100 features are extracted for classification and numeral recognition.

5. Evaluated classifiers

We classified numeral images in two stages. In first stage classification, above discussed features are fed to MLP's, designed separately for all four features for recognition. Results of four MLP's are combined using weighted majority scheme[12]. Numerals not classified by MLP, in first stage are classified using SVM, in second stage.

5.1 Neural Network

We used the same MLP with 3 layers including one hidden layer for four different feature sets consisting of 24 shadow features, 100 zone based centroid features, 200 zone based directional features and 44 view based features. The experimental results obtained while using these features for recognition of handwritten Devnagari numerals is presented in the next section. At this stage all numerals are non-compound, single numeral so no segmentation is required.

The classifier is trained with standard Backpropagation [9]. It minimizes the sum of squared errors for the training samples by conducting a gradient descent search in the weight space. As activation function we used sigmoid function. Learning rate and momentum term are set to 0.8 and 0.7 respectively. As activation function we used the sigmoid function. Numbers of neurons in input layer of MLPs are 24, 44, 200 or 100, for shadow features, view based features, zone based directional features and zone based centroid features respectively. Number of neurons in Hidden layer is not fixed, we experimented on the values between 20-70 to get optimal result and finally it was set to 30, 40, 30 and 70 for

shadow features, zone based centroid features, view based features and zone based directional features respectively. The output layer contained one node for each class., so the number of neurons in output layer is 10.

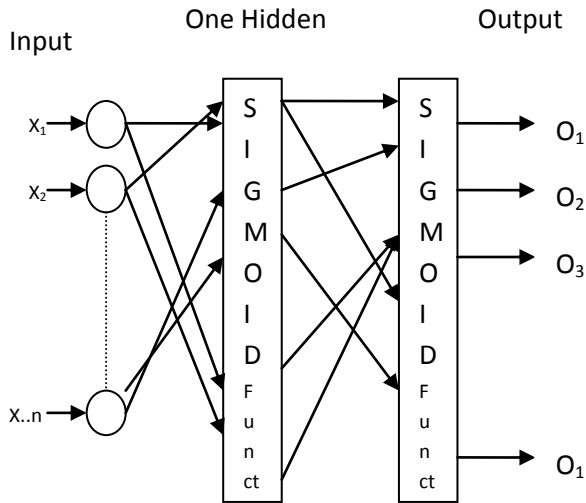


Figure 4. Block diagram of MLP

Outputs from several classifiers can be combined to produce a more accurate result. We have four similar Neural networks classifiers as discussed above, which are trained on 24 shadow features, 200 zone based directional based features, 44 view based features and 100 zone centroid based features respectively. The outputs are confidences associated with each class. As these outputs cannot be compared directly, we used an aggregation function for combining the results of all four classifiers. Our strategy is based on weighted majority voting scheme [12].

5.2 Support Vector Machines

The objective of any machine capable of learning is to achieve good generalization performance, given a finite amount of training data, by striking a balance between the goodness of fit attained on a given training dataset and the ability of the machine to achieve error-free recognition on other datasets. With this concept as the basis, support vector machines have proved to achieve good generalization performance with no prior knowledge of the data.

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. A special property of SVMs is that they simultaneously minimize the empirical classification

error and maximize the geometric margin; hence they are also known as maximum margin classifiers. Viewing the input data as two sets of vectors in an n dimensional space, an SVM will construct a separating hyper plane in that space, one which maximizes the "margin" between the two data sets. To calculate the margin, construct two parallel hyper planes, one on each side of the separating one, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the neighboring data points of both classes. The hope is that, the larger the margin or distance between these parallel hyper planes, the better the generalization error of the classifier will be. Consider training data set $\{(x_1, c_1), (x_2, c_2) \dots (x_n, c_n)\}$ where the c_i is either 1 or -1, indicating the class to which the point belongs. Each is a p -dimensional real vector. We want to give the maximal-margin hyper plane which divides the points having $c_i = 1$ from those having $c_i = -1$. Any hyper plane can be written as the set of points satisfying Maximum-margin hyper plane and margins for a SVM trained with samples from two classes. Samples on the margin are called the support vectors. Linear classifier is shown on the Figure 5.

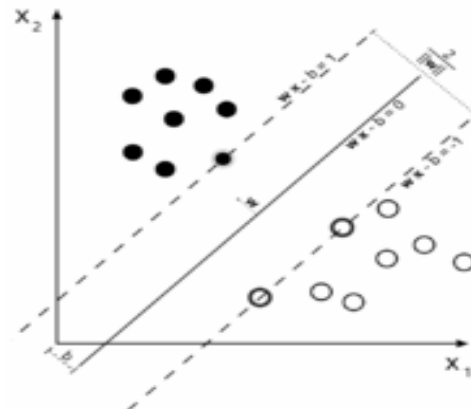


Figure 5. SVM with Linear Classifier

The points x lies on the hyper plane satisfy $w \cdot x + b = 0$, the vector w is a normal to the hyper plane. $|b|/||w||$ is a perpendicular distance from hyper plane to the origin as shown in Figure 9 and $||w||$ is the Euclidean norm of w . The parameter b determines the offset of the hyper plane from the origin along the normal vector. Choose the w and b to maximize the margin, or distance between the parallel hyper planes that are as far apart as possible while still separating the data. These hyper planes can be described by the equations

$$x_i \cdot W + b \geq +1 \quad \text{for } C_i = +1$$

$$x_i \cdot W + b \leq -1 \quad \text{for } C_i = -1$$

Note that if the training data are linearly separable, select the two hyper planes of the margin in a way that there are no points between them and then try to maximize their distance. By using geometry, we find the distance between these two hyper planes is $2/|w|$, so we want to minimize $|w|$. The optimal separating hyper plane can be determined without any computations in the higher dimensional feature space by using kernel functions in the input space. We used linear kernel $K(x, y) = x.y$, because they are simple and can be computed quickly. There is no kernel parameter choices needed to create a linear SVM, but it is necessary to choose a value for the soft margin in advance.

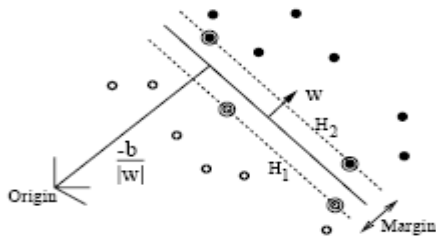


Figure 6. Linear separating hyper planes for the separable case

6. Performance Evaluation

The experiments of character recognition reported in the literature vary in many factors such as the sample data, pre-processing technique, feature representation, classifier structure and learning algorithm. Only a few works have applied different classification/learning methods for classification. We tested performance of Handwritten Devnagari Numerals using ANN and SVM applied at different stages. At first stage, numeral images are classified using four MLP's designed using four features (section 4). Results of different features using MLP's are given in Table 1. Results of four MLP's are combined using weighted majority scheme, illustrated in Table 2. Numerals not classified by MLP in first stage, are fed to SVM, in second stage for classification, which gives 96.29% accuracy. Results of both classifiers are combined to get higher accuracy. The overall accuracy achieved is 93.15%, by combining the 92.91% accuracy of MLP and 96.29% accuracy of SVM.

The experiment of Devnagari numerals dataset contains 18300 handwritten samples, 12810 samples in train dataset and 5490 samples for test results. The detailed recognition results of individual numerals are given in Table 4 and Table 5. In Table 4, results of MLP on individual numeral are given. Numerals not classified by MLP are fed to SVM. Table 5, gives the results of

individual numerals classified by SVM in second stage, i.e. numerals not classified by MLP in first stage. Detail comparisons of results are given in Table 3.

Table 1. Results of ANN for different features

MLP	Input layer Neuron	Hidden Layer Neuron	Output Layer Neuron	Result
Shadow based features	24	30	10	85.19%
Zone based Centroid features	100	40	10	80.87%
View Based Features	44	30	10	86.87%
Zone based directional Feature	200	70	10	90.92%

Table 2: Accuracy of ANN in first stage and SVM in second stage





















Classifier	Accuracy
ANN using weighted majority scheme	92.91% (top 1) 97.31% (top 2) 99.66% (top 3) 100% (top 4)
SVM	96.29%
Combined ANN (top 1) and SVM	93.15%

Table 3: Comparison of Results

Sl.No.	Method Proposed by	Technique	Data set	Accuracy
1	Hanmandlu and Ramana Murthy[4]	Fuzzy model based recognition using exponential membership function fitted to fuzzy sets derived from features consisting of normalized distances obtained using the Box approach.	350	95%
2	Bajaj al. [5]	density features, moment features and descriptive component features with multi-classifier connectionist architecture	2460	89.6%
3	Bhattacharaya et al. [6]	Multi-Layer Perceptron (MLP) neural network based classification approach on shape feature vector computed from certain directional-view-	22535	91.28%

		based strokes of numeral image		
4	C. V. Lakshmi, et al. [15]	Edge directions histograms and splines along with PCA	9800	94.25%
5.	R. J. Ramteke, S. C. Mehrotra[16]	central invariant moments as features with Gaussian Distribution Function for classification	2000	92%
6.	Our proposed method	Shadow, zone based directional,view based, zone based centroid features with MLP neural network and SVM based classification approach in two stages	18300	93.15%

Table 4: Individual numeral accuracy using ANN in first stage Table 5: Accuracy using SVM in second stage

Numeral	Accuracy	Numeral	Accuracy
	96.96%		100%
	96.96%		100%
	72.72%		66.66%
	87.87%		100%
	96.96%		100%
	87.87%		100%
	96.96%		100%
	93.93%		100%
	87.87%		100%
	93.93%		100%

7. Conclusion

The result obtained for recognition of Devnagari numerals show that reliable classification is possible using SVMs. We applied SVMs on different feature data namely Shadow based, Chain code Histogram, View based features and zone centroid based features. The SVM-based method described here for offline Devnagari

numerals can be easily extended to other Indian scripts numerals also.

References

- [1] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten character recognition: A comprehensive survey", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, pp 62-84, 2000.
- [2] Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Hubbard, L. D. Jackel and D. Henderson, "Handwritten digit recognition with a back-propagation network", Advances in neural information processing systems, 1990, pp 396 to 404.
- [3] I.K. Sethi and B. Chatterjee, "Machine Recognition of constrained Hand printed Devnagari", Pattern Recognition, Vol. 9, pp. 69-75, 1977.
- [4] M. Hanmandlu and O.V. Ramana Murthy, "Fuzzy Model Based Recognition of Handwritten Numerals", Pattern Recognition, 40 (1840-1854), 2006.
- [5] Reena Bajaj, Lipika Dey, and S. Chaudhury, "Devnagari numeral recognition by combining decision of multiple connectionist classifiers", Sadhana, Vol.27, part. 1, pp.-59-72, 2002.
- [6] U. Bhattacharya, B. B. Chaudhuri, R. Ghosh and M. Ghosh, "On Recognition of Handwritten Devnagari Numerals", In Proc. of the Workshop on Learning Algorithms for Pattern Recognition (in conjunction with the 18th Australian Joint Conference on Artificial Intelligence), Sydney, pp.1-7, 2005.
- [7] U. Pal and B.B. Chaudhuri, "Indian script character recognition: A Survey", Pattern Recognition, Vol. 37,pp. 1887-1899, 2004.
- [8] M. Tellache, M. A. Sid-Ahmed, B. Abaza . Thinning algorithms for Arabic OCR. IEEE Pac Rim 1993. 248-251
- [9] J. Hertz, A. Krogh, R.G. Palmer, "An Introduction to neural Computation", Addison-Wesley (1991)
- [10] S. Basu, N.Das, R. Sarkar, M. Kundu, M. Nasipuri, D.K. Basu, "Handwritten Bangla alphabet recognition using MLP based classifier", NCCPB, Bangladesh, 2005
- [11] S. Arora, D. Bhattacharjee, M. Nasipuri , D.K. Basu , M.Kundu , "Study of Different Features on Handwritten Devnagari Character",ICETET09, Nagpur ,India2009.
- [12] S. Arora, D. Bhattacharjee, M. Nasipuri , D.K. Basu , M.Kundu "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", 2008 IEEE Region 10 Colloquium and

the Third ICIIIS, IIT Kharagpur, INDIA December 8-10, 2008.

- [13] S. Arora , D. Bhattacharjee, M. Nasipuri , D.K. Basu , M.Kundu,” Application of Statistical Features in Handwritten Devnagari Character Recognition “, International Journal of Recent Trends in Engineering, Vol 2, No. 2, November 2009,pp40-42
- [14] S. Arora , D. Bhattacharjee, M. Nasipuri , D.K. Basu , M.Kundu ,”Recognition of Non-Compound Handwritten Devnagari Characters using a Combination of MLP and Minimum Edit Distance” , International Journal of Computer science and security vol 4 issue 1 :2010
- [15] C. Vasantha Lakshmi, Ritu Jain, C.Patvardhan, “Handwritten Devnagari Numerals Recognition with higher accuracy”, International Conference on Computational Intelligence and Multimedia Applications 2007
- [16] R. J. Ramteke, S. C. Mehrotra,”Recognition of Handwritten Devnagari Numerals”, International Journal of Computer Processing of Oriental Languages, 2008
- [17] S. Arora, D. Bhattacharjee, M. Nasipuri , L.Malik,“A Novel Approach for Handwritten Devanagari Character Recognition” in IEEE – International Conference on Signal And Image Processing, Hubli, Karnataka, Dec 7-9, 2006

Real Time Wireless Sensor Network for Coastal Erosion using Fuzzy Inference System

Arabinda Nanda*, Amiya Kumar Rath†, Saroj Kumar Rout

*: Department of Computer Science , Krupajal Engineering College, Bhubaneswar , India , Email: aru.nanda@rediffmail.com ,
rout_sarojkumar@yahoo.co.in

†: Department of Computer Science & IT , College of Engineering , Bhubaneswar, India , Email: amiyaamiya@rediffmail.com

Abstract—Wireless sensor networks (WSN) are one of the research areas in 21st century, which provide platform to scientists with the capability of developing real-time monitoring systems. This paper discusses the development of a WSN to detect coastal erosions, which includes the design, development and implementation of a WSN for real time monitoring, the development of the algorithms needed that will enable efficient data collection and data aggregation, and the network requirements of the deployed coastal erosions detection system. The actual deployment of Puri Sea Beach is in the Puri district of the state of Orissa, India, a region renowned for the sand sculptures and become a favorite haunt of both Indian and foreign beach lovers.

Keywords- *wireless sensor network, distributed algorithms, heterogeneous networks, coastal erosion.*

1. Introduction

India has a long coastline of 7516.6 km (according to National Hydrographic Office, Dehradun), spread along the nine maritime states of Orissa, Andhra Pradesh, West Bengal, Tamil Nadu, Kerala, Karnataka, Goa, Maharashtra, Gujarat and the Union Territories of Pondicherry, Andaman & Nicobar Islands, Lakshadweep Islands and Daman & Diu. A substantial portion of the country's coast is affected by sea erosion. The causes of coastal erosion can be natural and/or man-made [4].

Environmental disasters are largely unpredictable and occur within very short spans of time. Therefore technology has to be developed to capture relevant signals with minimum monitoring delay. Wireless sensors are one of the latest technologies that can quickly respond to rapid changes of data and send the sensed data to a data analysis center in areas where cabling is not possible.

WSN technology has the capability of quick capturing, processing, and transmission of critical data in real-time with high resolution. However, it has its own limitations such as relatively low amounts of battery power and low memory availability compared to many existing technologies. It does, though, have the advantage of deploying sensors in hostile environments with a bare

minimum of maintenance. This fulfills a very important need for any real time monitoring, especially in unsafe or remote scenarios.

We aim to use the WSN in the coastal erosion scenario for estimating the occurrence of erosions. In India, about 1,500 kilometers' or 26 % of the mainland coastline faces 'serious erosion' and is 'actively retreating', according to the Asian Development Bank. Coastal erosion is responsible for the loss of land, houses, infrastructure, and business opportunities and poses a high risk to human well-being, economic development, and ecological integrity. Coastal erosion has resulted in loss of life, property, valuable beaches and coastal land used for habitation, agriculture and recreation and continues to be a serious threat to many important buildings, factories, monuments of historical importance, highways and strategic installations along the country's coast. It affects negatively the livelihood of coastal communities, particularly poor households, and ultimately the coastal economies. The annual land losses due to coastal erosion in India is estimated at around \$127 million; potentially the impact could be much more extensive and widespread in the period ahead as the coastline is increasingly subject to a wide range of economic developments; many of which create conflicts and pressures on the already disturbed natural coastal environments.

This paper discusses the design and deployment of a erosion detection system using a WSN system at Puri beach, Puri (Dist), Orissa (State), India. The increase in depressions during the monsoons over Bay of Bengal is directly related to rise in the temperature of sea surface. It is an impact of global warming. Abnormal behavior of sea surface temperature has started to affect the atmospheric climate over the Bay of Bengal. The increased number of continuous depressions over the Bay of Bengal has also led to increase in the height and velocity of the sea waves, which causes more erosion on the sea coast.

The remainder of the paper is organized as follows. Section 2 describes Research Background and Related Work. In Section 3, we describe the Neural Network Algorithm. Section 4 Wireless Sensor Test Bed. Section 5 Conclusion and Future Work.

2. Research Background and Related Work

The research background and relevant technologies includes: (1) the definition of erosion, (2) wireless sensor network technology, and (3) the neural network algorithm

2.1 Definition of Coastal erosion

What is Coastal /Sea erosion?

The landward displacement of the shoreline caused by the forces of waves and currents is termed as *coastal erosion* [1].

Causes of Erosion?

Coastal erosion occurs when wind, waves and long shore currents move sand from the shore and deposits it somewhere else.

Major Causes of Coastal Erosion are:-

Natural Causes

- Action of Waves.
- Winds.
- Tides.
- Near-shore currents.
- Storms.
- Sea Level Rise

Anthropogenic Causes (Human intervention causes)

- dredging of tidal entrances
- Construction of harbors in near shore.
- Construction of groins and jetties
- River water regulation works
- Hardening of shorelines with seawalls.
- Construction of sediment-trapping upland dams
- Beach nourishment.
- Destruction of mangroves and other natural buffers
- Mining or water extraction

2.2 Wireless Sensor Network Technology

WSN technology has generated enthusiasm in computer scientists to learn and understand other domain areas which have helped them to propose or develop real time deployments. One of the major areas of focus is environmental monitoring, detection and prediction.

The Drought Forecast and Alert System (DFAS) has been proposed and developed in [3]; it uses mobile

communication to alert the users, whereas the deployed system uses real time data collection and transmission using the wireless sensor nodes, Wi-Fi, satellite network and also through internet. The real streaming of data through broadband connectivity provides connectivity to wider audience.

An experimental soil monitoring network using a WSN is presented in reference [2], which explores real-time measurements at temporal and spatial granularities.

In this paper, real time deployment of a heterogeneous network for coastal erosion detection has been discussed. This study incorporates both theoretical and practical knowledge from diverse domains such as coastal erosion and geomechanics, wireless sensor, Wi-Fi, and satellite networks, power saving solutions, and electronic interface and design, among others, which covered the design, development and deployment of a real-time coastal erosion system using a WSN.

3. Mamdani fuzzy model

There are 3 types of fuzzy control system/model used.

1. Mamdani Fuzzy model
2. Sugeno Fuzzy model
3. Tsukamoto Fuzzy model

The most commonly used fuzzy inference technique is the so-called **Mamdani** method. In 1975, Professor Ebrahim Mamdani of London University built one of the first fuzzy systems to control a steam engine and boiler combination. He applied a set of fuzzy rules supplied by experienced human operators. The Mamdani-style fuzzy inference process is performed in four steps:

1. Fuzzification of the input variables
2. Rule evaluation (inference)
3. Aggregation of the rule outputs
4. Defuzzification.

Step 1: Fuzzification

The first step is to take the crisp inputs, x_1 , y_1 and z_1 (depression over sea, temperature over sea and height & velocity of wave), and determine the degree to which these inputs belong to each of the appropriate fuzzy sets. We examine a simple three-input one-output problem that includes two rules:

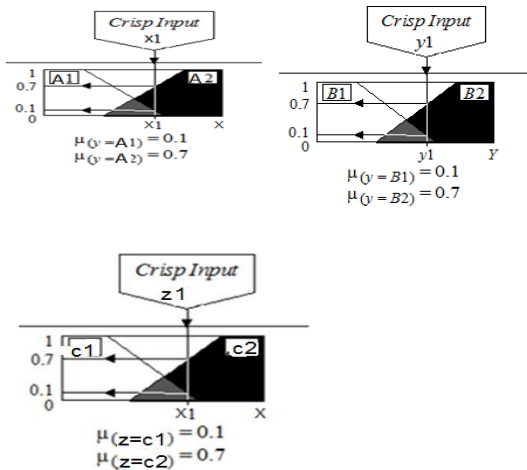
Rule: 1 IF x is A_2 AND y is B_2 THEN r is O_2

Rule: 2 IF x is A_2 AND z is C_2 THEN r is O_2

The Reality for these kinds of rules:

Rule: 1 IF depression over sea is more AND temperature over sea is more THEN erosion is more.

Rule: 2 IF depression over sea is more AND height, velocity of wave is more THEN erosion is more.



Step 2: Rule Evaluation

The second step is to take the fuzzified inputs, $\mu_{(x=A1)} = 0.1$, $\mu_{(x=A2)} = 0.7$, $\mu_{(y=B1)} = 0.1$, $\mu_{(y=B2)} = 0.7$ and $\mu_{(z=C1)} = 0.1$, $\mu_{(z=C2)} = 0.7$. Apply them to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation.

RECALL: To evaluate the disjunction of the rule antecedents, we use the **OR** fuzzy operation. Typically, fuzzy expert systems make use of the classical fuzzy operation union:

$$\mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)]$$

Similarly, in order to evaluate the conjunction of the rule antecedents, we apply the **AND** fuzzy operation intersection:

$$\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)]$$

Rule: 1 IF x is A2 (0.7) AND y is B2 (0.7) THEN r is O2 (0.7)

Rule: 2 IF x is A2 (0.7) AND z is C2 (0.7) THEN r is O2 (0.7)

Step 3: Aggregation of the Rule Outputs

Aggregation is the process of unification of the outputs of all rules. We take the membership functions of all rule

consequents previously clipped or scaled and combine them into a single fuzzy set. The input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable.

$$r \text{ is O2 (0.7)} \rightarrow r \text{ is O2 (0.7)} = \sum$$

Step 4: Defuzzification

The last step in the fuzzy inference process is defuzzification. Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number. There are several defuzzification methods, but probably the most popular one is the **centroid technique**. It finds the point where a vertical line would slice the aggregate set into two equal masses. Mathematically this **centre of gravity (COG)** can be expressed as:

$$COG = \frac{\sum_{x=a}^b x.m(x)}{\sum_{x=a}^b m(x)}$$

Centroid defuzzification method finds a point representing the centre of gravity of the aggregated fuzzy set A, on the interval [a, b]. A reasonable estimate can be obtained by calculating it over a sample of points. The final output of defuzzification will be the erosion degree.

4. Wireless Sensor Test Bed

The WSN follows a two-layer hierarchy, with lower layer wireless sensor nodes, sample and collect the heterogeneous data from the sensor column and the data packets are transmitted to the upper layer. The upper layer aggregates the data and forwards it to the sink node (gateway) kept at the deployment site. Data received at the gateway has to be transmitted to the Field Management Center (FMC) which is approximately 500mt away from the gateway. A Wi-Fi network is used between the gateway and FMC to establish the connection. The FMC incorporates facilities such as a VSAT satellite earth station and a broadband network for long distant data transmission. The VSAT satellite earth station is used for data transmission from the field deployment site at puri sea beach, Orissa, India to the Data Management Center (DMC), situated within the state.

The DMC consists of the database server and an analysis station, which performs data analysis and coastal erosion modeling and simulation on the field data to determine the erosion probability. The wireless sensor network architecture for coastal erosion detection is as shown in Fig-3.

The puri coastal region experiences frequent erosion and has erosion prone areas within 30 k.m (konark, puri_konark marine drive) which can be utilized as future extension sites for erosion detection systems. The different deployment sites can connect to the FMC via a Wi-Fi network.

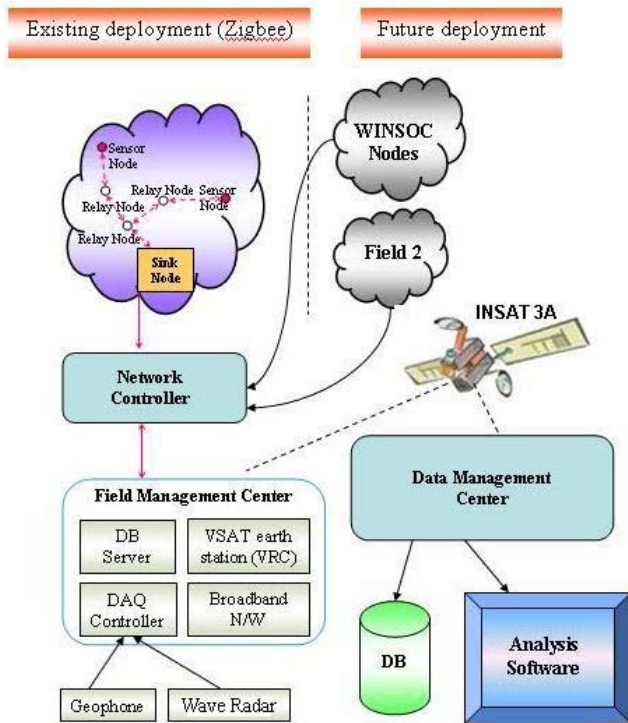


Fig-3 Wireless Sensor Network Architecture for Erosion Detection

5. Conclusion and Future Work

Real time monitoring of coastal erosion is one of the research areas available today in the field of geophysical research. This paper discusses the development of an actual field deployment of a WSN based coastal erosion detection system. This system uses a heterogeneous network composed of WSN, Wi-Fi, and satellite terminals for efficient delivery of real time data to the DMC, to enable sophisticated analysis of the data and to provide erosion warnings and risk assessments to the inhabitants of the region. In the future, this work will be extended to a full deployment by using the lessons learned from the existing network. This network will be used for understanding the capability and usability of WSN for critical and emergency

application. In the future, we plan to experiment with this method, including a simulation and implementation, to evaluate its performance and usability in a real sensor network application.

References

- [1] Coastal & Sea Erosion: Current Science, Vol-91, No-4, Aug 2006.
- [2] E. R. Musaloiu, A. Terzis, K. Szlavecz, A. Szalay, J. Cogan, and J. Gray, "Life under your feet: A wireless soil ecology sensor network", 2006
- [3] H. Kung, J. Hua, and C. Chen, "Drought forecast model and framework using wireless sensor networks, Journal of Information Science and Engineering, vol. 22, 2006, pp. 751-769
- [4] J. Chandrasekhar Iyer, "Workshop on coastal protection Measures, 5th & 6th Nov. 2004.

Authors Profile



Prof. Arabinda Nanda: Received M. Tech (CS) from Utkal University in the year 2007. Currently working as Assistant Professor in the Department of Computer Science & Engineering at Krupajal Engineering College, Bhubaneswar, Orissa, India. Contributed more than 10 research level papers to many National and International journals and conferences. Having research interests include Sensor Network, Adhoc Network, Soft Computing, Artificial Intelligence and Data Mining.



Prof (Dr) Amiya Kumar Rath: Obtained B.E. degree in Computer Science & Engg. from Marathwada University, Maharashtra in the year 1990, MBA degree in Systems Management from Shivaji University in the year 1993, M.Tech in Computer Science from Utkal University in year 2001 and Ph.D in Computer Science in the year 2005 from Utkal University for the work in the field of Embedded system. Served in various positions in different premier institutes namely College of Engineering, Osmanabad, Maharashtra, Orissa Engineering College, Bhubaneswar, Kalinga Institute of Industrial technology (KIIT), Bhubaneswar, Krupajal Engineering College, and Bhubaneswar in the Department CS & IT Engg. Presently working with College of Engineering Bhubaneswar (CEB) as

Professor of Computer Science & Engg. Cum Director (A&R) and is actively engaged in conducting Academic, Research and development programs in the field of Computer Science and IT Engg. Contributed more than 30 research level papers to many national and International journals. and conferences Besides this, published 4 books by reputed publishers. Having research interests include Embedded System, Adhoc Network, Sensor Network, Power Minimization, Biclustering, Evolutionary Computation and Data Mining.



Prof. Saroj Kumar Rout: Received M. Tech (CS) from Utkal University in the year 2007. Currently working as Assistant Professor in the Department of Computer Science and Engineering at Krupajal Engineering College, Bhubaneswar, Orissa, India. Contributed more than 08 research level papers to many National and International journals and conferences. Having research interests include Sensor Network, Adhoc Network, Embedded System, and Network Distributed System.

SVM Classifier Technique for Fingerprint Based Gender Identification

Arun K.S.¹, Sarath K.S.²

¹Department of Computer Science and Engineering, St. Joseph's College of Engineering, Kerala, India.
arunks11@yahoo.com

²Department of Electronics and Communication, College of Engineering Kidangoor, Kerala, India
iamsarathks@gmail.com

Abstract: Skin on human fingertips contains ridges and valleys which together forms distinctive patterns. These patterns are fully developed under pregnancy and are permanent throughout whole lifetime. Prints of those patterns are called fingerprints. Through various studies it has been observed that not two persons have the same fingerprints, hence they are unique for every individual. Fingerprints have remarkable permanency and individuality over the time. Above mentioned properties that human fingerprints have, made them very popular as biometrics measurements. Gender classification from fingerprints is an important step in forensic anthropology in order to identify the gender of a criminal and minimize the list of suspects search. The project deals with the problem of gender classification using fingerprint images. The project proposes a method for identifying the gender using fingerprint based on different features extracted. The relevant features to be extracted that distinguish the gender are ridge thickness to valley thickness ratio (RTVTR), and the ridge density. The extracted features are then used to train support vector machine classifier which can later predict the gender based on the extracted data.

Keywords: SVM, RTVTR, Ridge Density, Biometrics, Radial basis function.

1. Introduction

A successful gender classification approach can boost the performance of many other applications including face recognition and smart human-computer interfaces. Since the credentials can be lost, stolen or duplicated, token-based or knowledge-based approaches for personal identification were not secure [1]. On the other hand, biometrics is a science of verifying and establishing the identity of an individual through physiological features or behavioral characteristics that are unique to the individual and hence cannot be stolen, lost or misused. For successful human identification features used for biometric technique should satisfy certain properties such as they should be available to or within every individual, it should remain unchanged and available all the times and they should be extracted efficiently and accurately and measured quantitatively [2].

Among all the biometrics, fingerprint which is the reproduction of a fingertip epidermis, produced when a finger is pressed against a smooth surface, is the most established and well studied thing. Different studies prove that gender identification from fingerprint images is possible using the three relevant features, i.e. Ridge thickness to

Valley thickness ratio (RTVTR), Ridge density and the White lines count. Studies showed that the males have higher ridge count than the females [3]. It was shown that both males and females have higher rightward directional asymmetry in the ridge count with the asymmetry being higher in males than females and higher incidence of leftward asymmetry in females. The above mentioned features are then passed to SVM classifier.

SVM is a learning machine used as a tool for data classification, function approximation, etc, due to its generalization ability and has found success in many applications [4, 5]. Feature of SVM is that it minimizes the upper bound of generalization error through maximizing the margin between separating hyper plane and dataset. SVM has an extra advantage of automatic model selection in the sense that both the optimal number and locations of the basis functions are automatically obtained during training. The performance of SVM largely depends on the kernel [6].

The rest of this paper is organized as follows. Section 2, gives system overview. Section 3, gives a brief description about feature extraction algorithms. The details of SVM classifier used was described in section 4. The experimental results were shown in section 5. Conclusion and future enhancements were given in section 6.

2. System Overview

Figure.1 shows the methodology for gender identification using fingerprint images. Each of the 15 fingerprints of a subject is collected. Pre-processing techniques like binarization and enhancement are performed on the fingerprints in the next stage. The relevant features that can distinguish gender from fingerprints are extracted by applying different image processing techniques. The final step is to train the SVM classifier with the extracted known data values which later classify the unknown fingerprints as a male or female fingerprint.

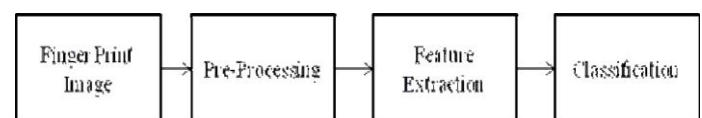


Figure 1. Block Diagram of the overall System

3. Feature Extraction

In pattern recognition and in image processing, Feature extraction is a special form of dimensionality reduction. Transforming the input data into the set of features is called features extraction. If the features extracted are carefully chosen it is expected that the feature set will contain relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

The two major features that are significant for gender classification using fingerprint images are:

- Ridge Thickness to Valley Thickness Ratio (RTVTR).
- Ridge Density.

3.1 Calculation of RTVTR

The RTVTR is the average ratio between the ridge thickness and valley thickness of a fingerprint. RTVTR is an important measure that can be used to classify the gender using fingerprint images. Female subjects are said to have higher RTVTR value compared to male subjects. The fingerprint image is divided into 32x32 non overlapping blocks. The local ridge orientation within each block is calculated. The projection profile of the valleys and ridges along a line perpendicular to the local ridge orientation in each block is calculated, and the projection profile was binarized using 1D optimal thresholding. The resultant binary profile represents the ridges and valleys in this block [7]. The average RTVTR is calculated for each block. For each block, a quality index was calculated as the average difference between the values of successive singular points (Minimas and Maximias) of the projection profile, blocks of good quality have higher quality index than those of bad quality.

3.1.1 Finding Local Ridge Orientation

The orientation image represents an intrinsic property of the fingerprint images and defines invariant coordinates for ridges and valleys in a local neighborhood. The quality of the ridge structures in a fingerprint image is an important characteristic, as the ridges carry the information of characteristic features required for minutiae extraction. Ideally, in a well-defined fingerprint image, the ridges and valleys should alternate and flow in locally constant direction. Thus, image enhancement techniques are often employed to reduce the noise and enhance the definition of ridges against valleys.

Normalization is used to standardize the intensity values in an image by adjusting the range of grey-level values so that it lies within a desired range of values [8]. Let $N(i, j)$ represent the normalized grey-level value at pixel (i, j) . The normalized image is defined as:

$$N(i, j) = \begin{cases} M_0 + \sqrt{\frac{VAR_0(I(i, j) - M)^2}{VAR}} & , \text{ if } I(i, j) > M \\ M_0 - \sqrt{\frac{VAR_0(I(i, j) - M)^2}{VAR}} & , \text{ Otherwise} \end{cases} \quad (1)$$

where M_0 and VAR are desired mean and variance respectively. M and VAR are mean and variance of the image respectively. The mean and variance of a gray-level fingerprint image I of size $N \times N$ are,

$$M(I) = \frac{1}{N^2} \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} I(i, j) \quad (2)$$

$$VAR(I) = \frac{1}{N^2} \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} (I(i, j) - M(I))^2 \quad (3)$$

Normalization is pixel-wise operation. It does not change the clarity of the ridge and furrow structures. The main purpose of normalization is to reduce the variation in gray level values along ridges and furrows, which facilitates the subsequent processing steps. Given a fingerprint image G , the main steps of the algorithm to find the local ridge orientation were as follows:

- Divide G into blocks of size $w \times w$ (32x32). Let the number of blocks be N .
- Compute the gradients $\delta_x(i, j)$ and $\delta_y(i, j)$ at each pixel (i, j) . The operator used is Sobel operator.
- Estimate the local orientation of each block centered at pixel (i, j) using:

$$\Delta_x(i, j) = \sum_{u=1}^{ww} \sum_{v=1}^{ww} 2\delta_x(u, v)\delta_y(u, v) \quad (4)$$

$$\Delta_y(i, j) = \sum_{u=1}^{ww} \sum_{v=1}^{ww} \delta_x(u, v)^2 - \delta_y(u, v)^2 \quad (5)$$

$$\Theta(i, j) = \tan^{-1} \frac{\Delta_y}{\Delta_x} \quad (6)$$

Where $\Theta(i, j)$ is the least square estimate of the local ridge orientation at the block centered at pixel (i, j) . Mathematically, it represents the direction that is orthogonal to the dominant direction of the fourier spectrum of the $w \times w$ window.

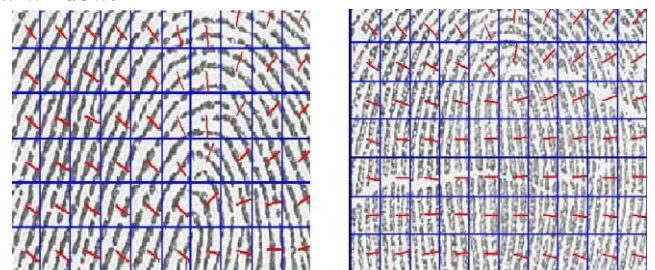


Figure 1. Ridge Orientation for male and female fingerprints

3.1.2 Binarizing the Fingerprint Image

Binarization is process where a grayscale image is decimated or categorized into only two levels, black and white (0 and 1). Since quality of fingerprints is varying, the grayscale image of the fingerprint will be more or less disturbed and noisy. Therefore binarization is performed in a way that it has enhancing effects on the fingerprint image.

Binarization of the fingerprint image is done as a preprocessing step before finding the projection profile. Thresholding is used for binarizing the image. Thresholding is the process of segmenting the image by scanning it pixel by pixel and labeling each pixel as object or background, depending on whether the gray level of that pixel is greater or less than the value of threshold T [9]. The following algorithm is used to obtain T automatically.

1. Select an initial estimate for T.
2. Segment the image using T. This will produce two groups of pixels: G1 consisting of all pixels with gray level values greater than T and G2 consisting of pixels with values $\leq T$.
3. Compute the average gray level values μ_1 and μ_2 for the pixels in regions G1 and G2.
4. Compute a new threshold value:

$$T \equiv \frac{1}{2} (\mu_1 + \mu_2) \quad (7)$$

5. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_0 .

The binarized fingerprint image is divided into non overlapping blocks of size 32 X 32. The projection profile for each of the block is obtained to find the RTVTR for the block.

3.1.3 Finding the Projection Profile

The horizontal projection profile is the histogram of the number of black pixels along every row of the image. The projection profile will have valleys of zero height between the ridges. The projection profile for all the blocks of the fingerprint is obtained as follows.

Each block is rotated by its local orientation angle. The projection profile of a block is calculated by counting the run of black pixels along the horizontal scan lines for the entire block. The projection profile is further binarized using 1D optimal thresholding to obtain the binary profile [10]. The zero height in the binary profile represent the valleys and the maximum height i.e. the peaks of the histogram represent the ridges. Figure 3 shows a fingerprint block and its projection profile.

The ridge and valley thickness is calculated by counting the consecutive zero values and maximum values respectively. The RTVTR for each block is given by:

$$RTVTR = \frac{\text{ridgethickness}}{\text{valleythickness}} \quad (8)$$

The obtained values for each block is averaged which gives the RTVTR value for the block. The RTVTR value for N blocks is averaged and this gives the RTVTR value for the fingerprint.

3.1.4 Finding the Projection Profile

The Optimal Thresholding [9] technique in one dimension is used for binarizing the projection profile. Figure. 4 shows a projection profile and its binary profile.

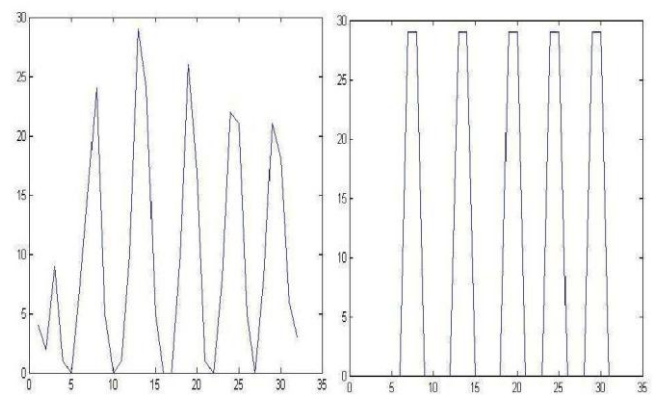


Figure 3. Projection profile and binary profile

The main steps in optimal thresholding technique for images is as follows [9]:

Algorithm

1. Consider as a first approximation that the four corners of the image contain background pixels only and the remainder contains object pixels.
2. At step t, compute μ_B^t and μ_O^t the mean background and object gray level, respectively, where segmentation into background and objects at step t is defined by the threshold value T^t determined in the previous step.

$$\mu_B^t = \frac{\sum_{(i,j) \in \text{background}} f(i,j)}{\# \text{ background_pixels}} \quad (9)$$

$$\mu_O^t = \frac{\sum_{(i,j) \in \text{Objects}} f(i,j)}{\# \text{ object_pixels}} \quad (10)$$

3. Set

$$T^{t+1} = \frac{\mu_B^t + \mu_O^t}{2} \quad (11)$$

provides an updated background object distinction.

4. If $T^{t+1} = T^t$ halt else return to step 2.

3.1.5 Finding the Quality Index for the Fingerprint Image Block

The uniformity of ridges and valleys within the blocks varies, for blocks having non uniform ridges and valleys due to the low quality of the fingerprint image in this region, the ridge orientation estimation is usually incorrectly estimated, and thus the RTVTR calculated for this block is incorrect, so only the blocks having the best quality should contribute to the average RTVTR calculated for this fingerprint [8]. For each block, a quality index was calculated as the average difference between the values of successive singular points (Minimas and Maximas) of the projection profile, blocks of good quality have higher quality index than those of bad quality. The blocks were arranged in a descending order based on their quality index, and the RTVTR of the best 20 were averaged and taken as the average RTVTR for the fingerprint. Figures 4 shows the projection profile of a good and bad block respectively.

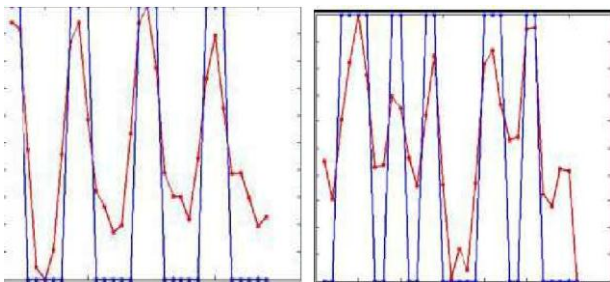


Figure 4. Profile of good and bad fingerprint blocks

3.2 Calculation of Ridge Density

The ridge density is calculated from the projection profile of the blocks that covers the upper portion of the fingerprint. The upper portion of each print was chosen as an area for the data collection because all fingerprint pattern types showed a similar ridge flow in this region [2]. The fingerprint image is enhanced. The enhancement is done in order to clearly define the ridges and valleys so that it will result in flawless counting of the ridges in the region of interest. The number of peaks in the projection profile of the top 128 x 128 block of a fingerprint image is counted, that gives the ridge density for the fingerprint. Fingerprint Enhancement The main purpose of image enhancement is to improve low quality images so that it can avoid the users to re provide their fingerprint just because it is not clearly impressed. A fingerprint image enhancement algorithm receives an input fingerprint image, applies a set of intermediate steps on the input image, and finally outputs the enhanced image. The algorithm for fingerprint image enhancement is given below [7,8].

- Normalization: An input fingerprint image is normalized so that it has a pre specified mean and variance.

- Local orientation estimation: The orientation image is estimated from the normalized input fingerprint image.
- Local frequency estimation: The frequency image is computed from the normalized input fingerprint image and the estimated orientation image.
- Region mask estimation: The region mask is obtained by classifying each block in the normalized input fingerprint image into a recoverable or a unrecoverable block.
- Filtering: A bank of Gabor filters which is tuned to local ridge orientation and ridge frequency is applied to the ridge-and-valley pixels in the normalized input fingerprint image to obtain an enhanced fingerprint image.

4. SVM Classifier

Consider the pattern classifier, which uses a hyper plane to separate two classes of patterns based on given examples $\{x(i), y(i)\}$. Where (i) is a vector in the input space $I = R^k$ and $y(i)$ denotes the class index taking value 1 or 0. A support vector machine is a machine learning method that classifies binary classes by finding and using a class boundary the hyper plane maximizing the margin in the given training data. The training data samples along the hyper planes near the class boundary are called support vectors, and the margin is the distance between the support vectors and the class boundary hyper planes. The SVM are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between assets of objects having different class memberships. SVM is a useful technique for data classification. A classification task usually involves with training and testing data which consists of some data instances [4,5]. Each instance in the training set contains one target value (class labels) and several attributes (features).

Given a training set of instance label pairs $(x_i, y_i), i=1 \dots l$ where $x_i \in R^n$ and $y_i \in \{1, -1\}$, the SVM require the solution of the following optimization problem:

$$\min_{w, b, \epsilon} \frac{1}{2} w^T w + c \sum_{i=1}^l \epsilon_i \quad (12)$$

subject to

$$y_i (w^T \phi(x_i) + b) > 1 - \epsilon_i, \quad (13)$$

$$\epsilon_i > 0 \quad (14)$$

where C is the capacity constant, w is the vector of coefficients, b a constant. Here training vectors x_i are mapped into a higher dimensional space by the function ϕ called the kernel. Then SVM finds a linear separating hyper plane with the maximal margin in this higher dimensional space. Thus support vector machines are an innovative approach to constructing learning machines that minimize

the generalization error. They are constructed by locating a set of planes that separate two or more classes of data. By construction of these planes, the SVM discovers the boundaries between the input classes; the elements of the input data that define these boundaries are called support vectors.

5. Experimental Results

The two main features were extracted from a dataset of 30 fingerprints containing 15 male subjects and 15 female subjects. Tables 1 brief the results. The results were shown graphically in figures 5.

Sl. No	RTVTR	Ridge Density	Sl.No	RTVTR	Ridge Density
1	1.2764	9	1	1.3006	11
2	1.0532	8	2	1.6425	11
3	1.4347	8	3	1.3952	12
4	1.2231	9	4	1.6164	10
5	0.9176	10	5	1.3998	12
6	1.2364	9	6	1.3990	12
7	1.1956	8	7	1.3983	13
8	1.2145	11	8	1.3846	12
9	1.0953	10	9	1.4523	9
10	1.2287	13	10	1.7834	10
11	1.3245	11	11	1.4849	11
12	1.1183	12	12	1.6689	12
13	0.8808	9	13	1.5934	11
14	1.2743	9	14	1.3207	9
15	1.2419	10	15	1.4264	13

Table 1. Feature Values for 15 male and female subjects

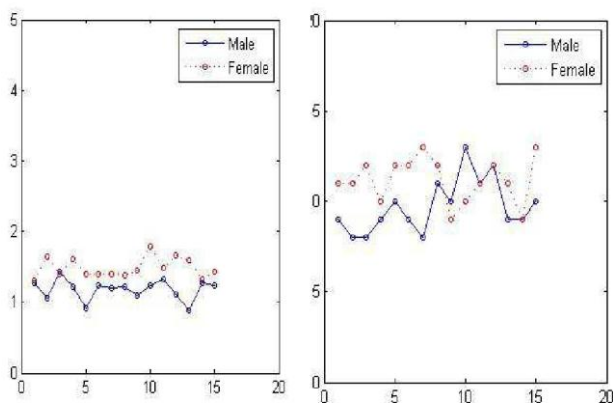


Figure 5. Plot of RTVTR and Ridge Density Values For 15 male and female subjects

From the experimental results it was clear that all the features extracted have higher values in female fingerprints than in male fingerprints. Gender classification results using these dominant features showed that this method could be considered as a prime candidate for use in forensic anthropology in order to minimize the suspects search list and give a likelihood probability value of the gender of a suspect. These extracted features were used to train the SVM classifier with the known result of male or female fingerprints. Further experiments showed that when a new dataset came the SVM classifier can predict the output with an accuracy of 89%.

6. Conclusions and Future Enhancements

This work proposes an SVM based gender classification system using fingerprints. The experimental results indicated that the females fingerprint is characterized by a high RTVTR, while the males fingerprint is characterized by low RTVTR, with the exception of small percentage of males fingerprints having high RTVTR, and females fingerprints having low RTVTR. The study also suggests that females have higher ridge density than males.

The future work is to incorporate additional features such as white line counts in the training as well as classification process and compare the performance of the new system with the existing one.

References

- [1] Ahmed Badawi, Mohamed Mahfouz, Rimon Tadross, Richard Jantz, "Fingerprint based Gender Classification", Proceedings of the 2006 International Conference on Image Processing, Computer Vision, & Pattern Recognition, Las Vegas, Nevada, USA, Volume 1; pp 41-46, June 26-29, 2006.
- [2] Dr. Sudesh Gungadin MBBS, MD, "Sex Determination from fingerprint ridge density", Internet Journal of Medical Update, Vol. 2, No. 2, Jul-Dec 2007.
- [3] Kralik M., Novotny V, " Epidermal ridge breadth : an indicator of age and sex in paleodermatoglyphics", Variability and Evolution, Vol. 11: 530, 2003.
- [4] C. Cortes, V. N. Vapnik, "Support vector networks", Machine learning Boston, vol.3, Pg.273-297, September 1995.
- [5] N. Acir, "A support vector machine classifier algorithm based on a perturbation method and its application to ECG beat recognition systems" Expert systems with application New York, vol.31, pg. 150-158 July 2006.
- [6] V. N. Vapnik, "An overview of statistical learning theory", IEEE Trans. Neural Networks New York, Vol. 10, pg. 998-999, September 1999.
- [7] L. Hong, Y.Wang, and A. K. Jain. "Fingerprint image enhancement: Algorithm and performance evaluation." , Transactions on PAMI, 21(4):777789, August 1998.

[8] L. Hong, A.K. Jain, S. Pankanti, and R. Bolle, "Fingerprint Enhancement," Proc. First IEEE WACV, p p. 202-207, Sarasota, Fla.,1996.

[9] S. Chikkerur and V. Govindaraju, "Fingerprint Image Enhancement Using STFT Analysis ",International Workshop on Pattern Recognition for Crime Prevention, Security and Surveillance (ICAPR 05), pages 20-29,2005.

[10] Jayadevan R., Jayant V. Kulkarni, Suresh N. Mali,Hemant K. Abhyankar, "A New Ridge Orientation based method for feature extraction from fingerprint images,"Proceedings of World Academy of Science , Engineering and Technology ,Volume 13 May 2006 ISSN 1307-6884

Author Biographies

Arun K.S. is a lecturer in Computer Science and Engineering at St. Joseph's College of Engineering, Kerala, India. He received his M-Tech in Computer Vision and Image Processing from Amrita School of Engineering Tamilnadu, India. He received his B-Tech in Computer Science and Engineering from MG University College of Engineering, Kerala, India. He served Tata Consultancy Service as an Assistant System Engineer. His current research interests include Medical Image Processing and Biometrics.

Sarath K.S. is an undergraduate student in College of Engineering Kidangoor, Kerala, India. His research interests includes Medical Image processing and Pattern Recognition

PERFORMANCE ANALYSIS OF LOSSLESS COMPRESSION SCHEMES FOR BAYER PATTERN COLOR IMAGE VIDEO SEQUENCES

G. Mohanbabu¹ and Dr. P. Renuga²

¹Lecturer, Department of Electronics and Communication Engineering, PSNACET, Dindigul
¹kgbabu73@yahoo.com

²Assistant Professor, Department of Electrical and Electronics Engineering, TCE, Madurai
²preee@tce.edu

Abstract: Most consumer digital color cameras are equipped with a single chip. Such cameras capture only one color component per pixel (e.g., Bayer pattern) instead of an RGB triple. Conventionally, missing color components at each pixel are interpolated from its neighboring pixels, so that full color images are constructed. This process is typically referred to as demosaicing. After demosaicing, the full resolution RGB video frames are converted into YUV color space. U and V are then typically subsampled by a factor of four and the resulting video data in the 4:2:0 format become the input for the video encoder. In this letter, we look into the weakness of the conventional scheme and propose a novel solution for compressing Bayer pattern video data. The novelty of our work lies largely in the chroma subsampling. We properly choose the locations to calculate the chroma pixels U and V according to the positions of B and R pixels in the Bayer pattern and this leads to higher quality of the reconstructed images. In our experiments, we have observed an improvement in composite peak-signal-to-noise ratio performance of up to 1.5 dB at the same encoding rate. Based on this highly efficient approach, we propose also a low-complexity method which saves almost half of the computation at the expense of a small loss in coding efficiency.

Keywords :- Bayer-pattern video compression, chroma subsampling, color space conversion, H.264/AVC.

1. Introduction

In recent years, digital cameras for still images and movies became popular. There are many obvious advantages to digital images comparing to classical film based cameras, yet there are limitations as well. For example, the spatial resolution is limited due to the physical structure of the sensors. "Superresolution" beyond the sensors resolution can be achieved by considering a sequence of images. To reduce cost, most digital cameras use a single image sensor to capture color images. A Bayer color filter array (CFA), as shown in Fig. 1, is usually coated over the sensor in these cameras to record only one of the three color components at each pixel location. The resultant image is referred to as a CFA image in this paper hereafter. In general, a CFA image

is first interpolated via a demosaicing process to form a full color image before being compressed for storage. Fig. 2(a) shows the workflow of this imaging chain.

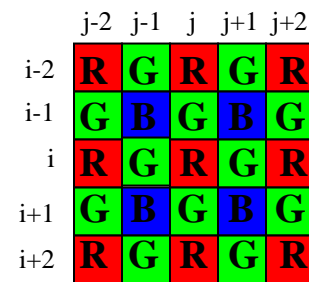


Fig.1. Bayer pattern having a red sample as its centre

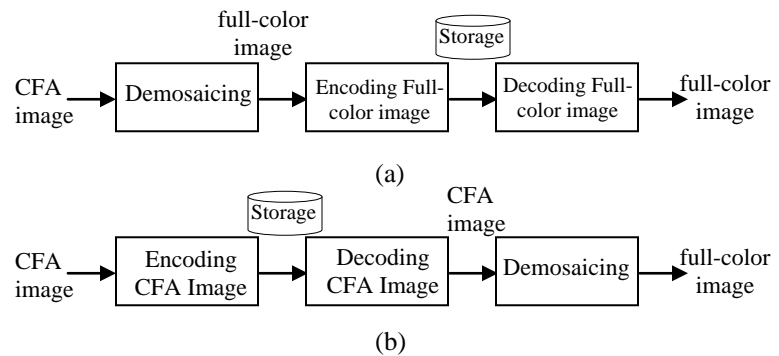


Fig.2. Single-sensor camera imaging chain: (a) the demosaicing- first scheme; (b) the compression- first

There are two categories of CFA image compression schemes: lossy and lossless. Lossy schemes compress a CFA image by discarding its visually redundant information. These schemes usually yield a higher compression ratio as compared with the lossless schemes. There are Some different lossy compression techniques such as discrete cosine transform, vector quantization, subband coding with symmetric short kernel filters, transform followed by JPEG or JPEG 2000, and low-pass filtering followed by JPEG-LS

or JPEG 2000 (lossless mode) are used to reduce data redundancy.

The red, green, and blue pixels in the Bayer pattern are separated into three arrays and then an MPEG-2 like video coder is used for compression. This method should have a limited coding efficiency for P-frames because severe aliasing is generally contained in Bayer pattern. To alleviate aliasing, it is proposed in [3], [4] that Bayer-pattern videos are compressed using an H.264 video coder and the motion compensation is adapted to the Bayer pattern. However, these two schemes are both confined to the RGB domain and, due partly to this, outperform the conventional method only at relatively high-bit rates.

In this letter, we propose a novel scheme using adjusted chroma subsampling for compressing Bayer-pattern video sequences. We first transform the Bayer-pattern images from the RGB domain to the YUV domain. In our approach, however, the chroma pixels U and V are calculated at different positions according to the B and R components of the Bayer pattern. Then we forward the image data in the YUV domain to an H.264 video coder for compression. After we have the YUV data decoded, we transform them back to the RGB domain so that the original Bayer-pattern image is reconstructed. Our proposed scheme proves significantly more efficient than the conventional one over the entire bit rate range. Moreover, this method can be extended to a low-complexity version. The computational complexity is reduced by almost 50% at the expense of a small drop in rate-distortion performance. In Section II, we describe the conventional approach and look into its drawback. In Section III, we present our proposed method in detail. The experimental results, including rate-distortion curves and encoding time, are presented in Section IV. Finally, Section V concludes the letter.

2. Conventional Approach

The Conventional Bayer-pattern video compression begins with demosaicing or color interpolation, as illustrated in Fig. 2(a). Then full color RGB images are converted to the YUV domain. After this, chroma subsampling is applied to the components U and V by a factor of two both horizontally and vertically, as shown in Fig. 1, so that the YUV data are available in the standard format 4:2:0. Now the YUV data are compressed using an H.264 video coder. The decoder reconstructs the YUV data in the format 4:2:0 and then interpolates the components U and V to their full size. Finally, the YUV images are converted back to RGB images.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.813 & 0.391 \\ 1.164 & 2.018 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2)$$

Simplicity is the most significant advantage of the conventional approach. There are existing or standard methods for all the main steps, including demosaicing, color space transform, chroma subsampling, and H.264/AVC video coding. However, such a simple combination of different techniques limits the coding efficiency. particular drawback lies in the chroma sample positions for chroma subsampling.

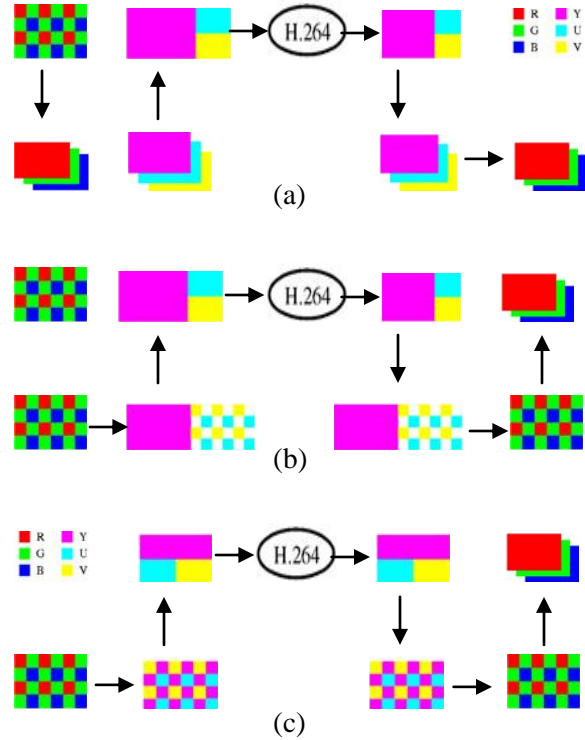


Fig. 2. Comparison of the conventional method, the proposed methods B-4:2:0 and B-4:2:2. (a) Conventional method 4:2:0. (b) Proposed method B-4:2:0. (c) Proposed method B-4:2:2.

3. Proposed Approach

A. Proposed Method B-4:2:0

The color space transform of our proposed method B-4:2:0 is illustrated in Fig. 3(a). We calculate Y pixels at all the locations in the Bayer-pattern image, no matter if it is an R pixel, a G pixel or a B pixel. When it comes to chroma pixels U and V, the positions to calculate them are carefully chosen. V values are calculated at the positions of R pixels, and U values at B pixels. We can justify this if we look into the equations in (2). During the inverse color space transform, only Y and V are needed to reconstruct R pixels. This means, V is more important than U at positions of R

pixels, that's why we calculate V values. For the same reason, U values are calculated where B pixels exist. The different positions selected for U and V pixels in our proposed scheme are more reasonable for the reconstruction of R and B pixels than the standardized chroma sample positions in the conventional method. The standard chroma subsampling takes U and V always at the same location without taking into account the different positions of R and B pixels in the Bayer pattern, thus it cannot be optimum for Bayer-pattern image and video compression. This is the fundamental reason, why our proposed scheme B-4:2:0 can have a better rate-distortion performance than the conventional one. Color space transform requires all three color components R, G, and B for every pixel, but Bayer-pattern images have only one, either R or G or B, at each position, so missing components for every pixel need to be interpolated from adjacent pixels. This process is just the so-called demosaicing or color interpolation. Now we base our discussion on demosaicing using bilinear interpolation and the equations are listed in Fig. 3(a). More advanced interpolation techniques can also be introduced into our system. In the section for experimental results, not only the results for bilinear interpolation are shown but also those for the edge-directed interpolation with secondorder gradients as correction terms, also known as Laplacian interpolation. The luma and chroma pixels are then written into a standard YUV file and forwarded to the H.264 video coder for compression. As shown in Fig. 3(a), for every 2×2 image block, all the four Y pixels are calculated and only one U pixel and one V pixel are calculated. Although the chroma value location differs from the nominal locations shown in Fig. 1, the resulting YUV images can be compressed using the 4:2:0 mode of H.264. At the decoder, YUV image data are first reconstructed. Of course, we also need to interpolate the missing U or V pixels in order to transform the data back to RGB Bayer-pattern images. Finally, we apply demosaicing by bilinear interpolation on the Bayer-pattern images to obtain RGB full color images.

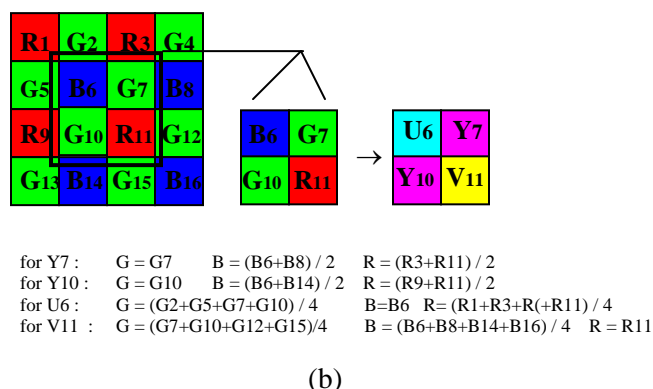
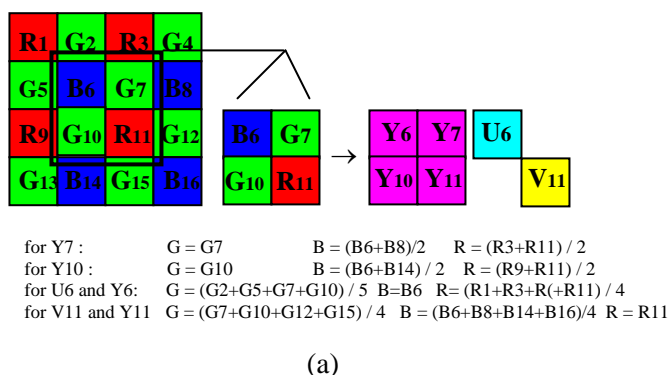


Fig. 3. Color space transform from the RGB domain to the YUV domain.

(a) Proposed method B-4:2:0. (b) Proposed method B-4:2:2.

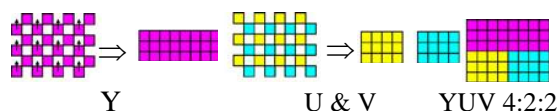


Fig. 4. Structure conversion for YUV data in the proposed method B-4:2:2

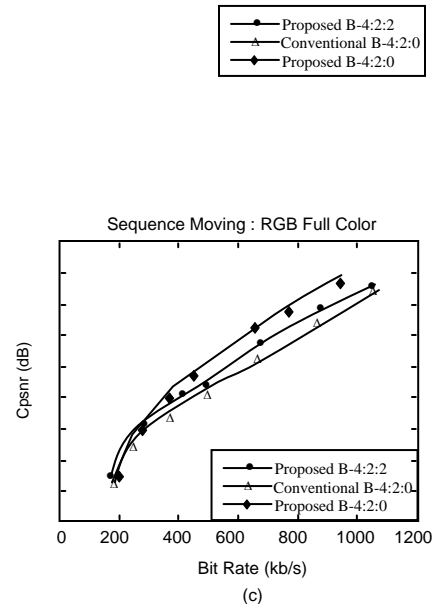
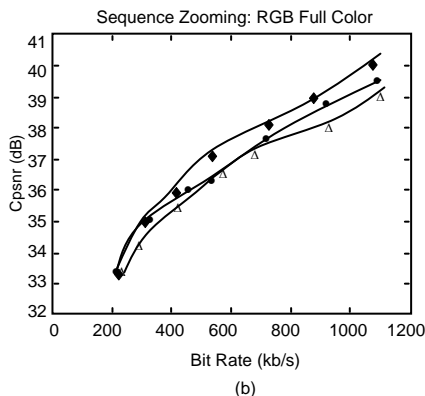
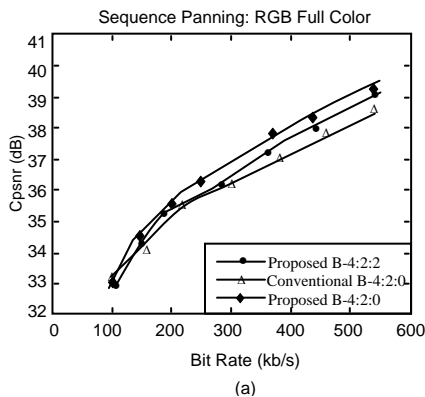
B. Proposed Method B-4:2:2

Based on the proposed method B-4:2:0, we propose the scheme B-4:2:2 mainly to further reduce the computational complexity. Instead of calculating all the luma pixels and compressing them, we keep only the Y pixels at the positions of G pixels and get rid of those at R and B pixels. Now that the Y pixels to compress are halved, the encoding time can be reduced almost by 50%. The color space transform of this method based on demosaicing using bilinear interpolation is illustrated in Fig. 3(b). The Y pixels we compress form a quincunx pattern as G pixels in the Bayer pattern. Therefore, it is necessary to add in a step of structure conversion [5], which converts this quincunx pattern to a rectangular one, in order that the H.264 video coder works correctly. Such structure conversion is shown in Fig. 4. We move Y pixels in the even rows one unit upward and then push the resulting complete rows of Y pixels together to a rectangular array. The chroma pixels are also pressed together. After this structure conversion, the YUV data match the standard format 4:2:2 and are ready to be compressed by an H.264 video coder using the 4:2:2 mode. The decoder needs to convert the rectangular pattern of the reconstructed Y pixels back to the quincunx pattern. Then missing Y pixels are interpolated before RGB values in the

Bayer pattern can be calculated. Finally, full color RGB images are generated by demosaicing the Bayer-pattern images. One thing worth mentioning is that the interpolation scheme we use for Y pixels is of the same order as the one for G pixels in the demosaicing of RGB images.

4. Experimental Results

For the simulation, we capture in our laboratory three Bayerpattern video sequences which represent three different motion modes, *Panning*, *Zooming*, and *Moving Object*, over a static background. The video frames are captured in common intermediate format (CIF) 352×288 and at a rate of 30 frames/s. Screenshots of selected frames of the three test videos are shown in Fig. 5. The H.264/AVC reference software JM 12.2 is used for video compression in our simulation. The group of picture (GOP) size is 40 and the GOP structure is set to I-P-P-P. The YUV format is set to 4:2:0 or 4:2:2, according to different simulations. As for other parameters in the configuration file of the JM reference software, we keep their default values. Rate-distortion curves for different methods and different test sequences are plotted in Fig. 6 for the case of bilinear interpolation. The original Bayer-pattern images and the reconstructed ones are interpolated to full color RGB images using bilinear interpolation and the composite peak-signal-to noise ratio (CPSNR) is calculated between them. Finally, the CPSNR for all the images in a sequence is averaged. Equations (3) and (4) are used to calculate CPSNR of an image.



$$CPSNR = 10 \log_{10} \frac{255^2}{MSE}$$

$$MSE = \frac{1}{3MN} \sum_{k=1}^3 \sum_{i=1}^N \sum_{j=1}^B [I(I,j,k) - \hat{I}(i,j,k)]^2$$

5. Conclusion

The rate-distortion performance of our proposed method B-4:2:2, however, could be further improved. As discussed, the structure conversion in the proposed method B-4:2:2 destroys the regular arrangement of Y pixels, resulting in a lower accuracy of motion compensated prediction in video coding. If we modify the motion estimation and compensation of the H.264 video coder, the coding efficiency of the method B-4:2:2 can become higher.

References

[1] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.

- [2] F. Gastaldi, C. C. Koh, M. Carli, A. Neri, and S. K. Mitra, "Compression of videos captured via bayer patterned color filter arrays," in *Proc. 13th Eur. Signal Process. Conf.*, Antalya, Turkey, Sep. 2005.
- [3] C. Doutre and P. Nasiopoulos, "An efficient compression scheme for colour filter array video sequences," in *Proc. IEEE 8th Workshop Multimedia Signal Process.*, Victoria, BC, Oct. 2006, pp. 166–169.
- [4] C. Doutre, P. Nasiopoulos, and K. N. Plataniotis, "H.264-based compression of bayer pattern video sequences," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 18, no. 6, pp. 725–734, Jun. 2008.
- [5] C. C. Koh, J. Mukherjee, and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1448–1456, Nov. 2003.
- [6] J. Mukherjee, M. K. Lang, and S. K. Mitra, "Color demosaicing in YUV color space," in *Proc. Int. Assoc. Sci. Tech. Dev. Conf.*, Malaga, Spain, Sep. 2002, pp. 96–101.
- [7] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.
- [8] S. Pei and I. K. Tam, "Effective color interpolation in CCD color filter array using signal correlation," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, Sep. 2000, pp. 10–13.
- [9] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 4, May 2002, pp. 3333–3336.
- [10] T. Kuno and H. Sugiura, "New interpolation method using discriminated color correlation for digital still cameras," *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 259–267, Feb. 1999.
- [11] C. Weerasinghe, I. Kharitonenko, and P. Ogunbona, "Method of color interpolation in a single sensor color camera using green channel separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 4, May 2002, pp. 3233–3236.
- [12] S. Ferradans, M. Bertalmio, and V. Caselles, "Geometry-based demosaicking," *IEEE Trans. Image Process.*, vol. 18, no. 3, pp. 665–670, Mar. 2009.
- [13] J. E. Adams and J. F. Hamilton, "Design of practical color filter array interpolation algorithms for digital cameras," in *Proc. Soc. Photo-Optical Instrum. Eng.*, vol. 3028, Feb. 1997, pp. 117–125.
- [14] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 44–54, Jan. 2005.
- [15] K. Jack, "Color spaces," in *Video Demystified*, 3rd ed. Eagle Rock, VA: LLH Technology Publishing, 2001, pp. 18–19.

A Framework for Semantic Web Services Discovery using Improved SAWSDL-MX

*Agushaka J. O and Junaidu S. B

Department of Mathematics, Ahmadu Bello University Zaria-Nigeria

*jagushaka@yahoo.com and sahalu@abu.edu.ng

Abstract - The increasing growth in the popularity of Web Services makes discovery of relevant Web Services a significant challenge. To this end, the central objective of this paper is to propose a framework that proffers solutions to some problems identified in the course of study of SAWSDL-MX, a hybrid matchmaker based on OWL-S MX, that uses both logic based reasoning and content based information retrieval techniques for services specified in SAWSDL description. As part of solution to the problems identified, non-functional specification like quality of service (QoS) was added to the discovery process and also, users are allowed to specify queries for services using terms in their own ontology. This is achieved by specifying a mapping from user ontology to domain ontology and also a QoS algorithm was developed and incorporated into the architecture of SAWSDL-MX to enable service discovery based on non-functional specification. An example is given in this paper to validate our framework. The implementation is an ongoing work.

Keywords: Semantic Web Services Discovery, SAWSDL-MX, OWL-S MX, Non-Functional Specification (QoS)

1. Introduction

Web applications are applications that are accessed via web browsers over a network such as the Internet or an intranet. Web has gone through many transformations starting from the first generation non-interactive content pages to the new generation Web Services. Web Services are the third generation web applications; they are modular, self-describing, self-contained applications that are accessible over the Internet Cubera et al (2001). In its simplest form, a Web Services is a class whose methods are callable remotely. Method calls to and responses from Web Services are transmitted via SOAP. Thus any client capable of generating and processing SOAP messages can use a Web Services regardless of the language in which the Web Services is written. Once a Web Service is deployed, other application (and other Web Services) can discover and invoke the deployed service. The major issue with the current Web Services is **Description**. Currently, WSDL contains information useful to humans not computer/applications i.e. the description lacks explicit semantic and this can lead to misinterpretation of meaning of description. As such, discovery which is a process of finding Web Services that matches request can't be automated. Currently, UDDI supports Web Services discovery based on keyword search and taxonomy based search which can be less effective than desired. The popularity of Web Services solution led to the proliferation of Web Services thereby making the keyword based matching of requests for services inadequate. Semantic Web technology gave hope to this problem however, the use of ontology to add meaning to Web Services description came with its own problem (ontology heterogeneity). Use of common ontology may solve this problem but individual users or communities of users are expected to query for services of interest to them using descriptions that are expressed using terms in their own ontologies. The need to

specify a mechanism for mapping user ontology to the domain ontology used to describe services is essential. Also, the need to add non-functional specification to the discovery process cannot be overemphasized. Consider a scenario where a request for service satisfies all the users' requirements (functional) but the cost (non-functional) of invoking the service is overbearing. If non-functional specification is not included in the discovery process, this very service will be returned to the user who will unknowingly invoke and bear the consequences. These amongst others motivated this research. Semantic Web Services Discovery is based on matching semantically described goal descriptions (goal queries) with semantic annotations of Web Services (capability descriptions) (Walsh et al, 2002). Several capability-based semantic Web Services discovery solutions have been proposed in the literature (Patil et al, 2003; Duftler et al, 2001; Cardoso et al, 2002; DAML-S 0.7 Draft Release, 2002). A capability description annotates either the inputs or outputs of Web Services (Cardoso et al, 2002) or describes an abstract service capability (Paolucci et al, 2002; Duftler et al, 2001) and can be applied in the frame of both OWL-S (Patil et al, 2003) and WSMF/WSMO (Ankolenkar et al, 2002; Paolucci et al, 2002). The discovery process in the capability-based semantic discovery approaches can only happen on an ontological level, i.e., it can only rely on conceptual and reusable things. The greatest difficulty in a Web Services discovery mechanism is *heterogeneity* between services (Garofalakis et al, 2004). Heterogeneities include different operating platforms, different data formats, as well as heterogeneities of ontologies. Regarding ontology heterogeneities, semantic Web Services may use different ontologies or different ontologies description language such as OWL, DAML, RDF, and so forth to describe the services. There is also heterogeneity between semantic Web Services and non-semantic Web Services. Therefore, when developing a discovery system, these heterogeneities should

be borne in mind. UDDI service registry does not allow, as it is, to store any semantic information related to service declarations, as such most semantic Web Service discovery engines allows for services to be registered in their service database and only services that are registered with the engine are considered during matching. However, in the work of Pierre (2007), a WSDL-S to UDDI mapping was defined and a service publication and querying API named LUCAS (layer for UDDI compatibility with annotated semantics) is placed around it. Jyotishman et al (2005) proposes a framework for ontology-based flexible discovery of Semantic Web Services. The proposed approach relies on user-supplied, context-specific mappings from a user ontology to relevant domain ontologies used to specify Web Services. A description of how user-specified preferences for Web Services in terms of non-functional requirements (e.g., QoS) can be incorporated into the Web Services discovery mechanism to generate a partially ordered list of services that meet user-specified functional requirements. OWL-S/UDDI (Srinivasan et al, 2004)) matchmaker combines UDDI's proliferation into the Web Services infrastructure and OWL-S's explicit semantic description of the Web Services. Glue (www.swa.cefriel.it/Glue) is a WSMO compliant discovery engine that aims at developing an efficient system for the management of semantically described Web Services and their discovery. OWL-S MX by Matthias et al (2008) is a hybrid OWL-S semantic service matchmaking algorithm. It uses both logical based and content based retrieval techniques for Web Services discovery. The hybrid semantic service matching uses six different filters to calculate the degree of semantic match between the request and advertisement. The first four filters are purely logic based and the next two are hybrid, which use the IR similarity metric values. SAWSDL –MX by Kapahnke et al (2008) is an approach that does hybrid semantic Web Services discovery using SAWSDL based on logic based matching as well as information retrieval based techniques. It is significantly based on but a further refinement of OWL-MX. Our work seeks to extend the work of Kapahnke et al. In the next section, we give some of the problems we identified in the cause of our study of SAWSDL-MX and then proffer solutions to them and that serves as our extension of Kapahnke's approach. We extended SAWSDL-MX to allow user to specifying queries using descriptions expressed using terms in their own ontology. This is to be achieved by specifying mapping between user ontology to service ontology. We also included non functional specification e.g. quality of service in the discovery process of SAWSDL-MX. It is important to note that the implementation of this work is ongoing and that this is a framework that is validated using the example here given. The rest of the paper is organized as follows: service matching using SAWSDL-MX (that we extended) is given in section 2, section 3 gives our proposed framework. Architecture of the framework is in section 4, an example of service discovery using our framework is in section 5, future work in section 6, section 7 is related work and finally conclusion and acknowledgements is given.

2. Service Matching with SAWSDL-MX

SAWSDL-MX was developed based on some assumptions. Details of which can be found in Kapahnke et al (2008). SAWSDL-MX performs logic-based, syntactic (text similarity-based) and hybrid matching of service against request. The request for service is given as a standard SAWSDL document. This approach is particularly based on OWLS-MX (Matthias et al, 2008) and WSMO-MX (Kaufer et al, 2006) for OWL-S and WSML respectively.

2.1. Logic-Based Operation Matching

SAWSDL-MX applies four matching filters of increasing degree of relaxation: Exact, Plug-in, Subsumes and Subsumed-by, which are adopted from OWLS-MX but modified in terms of an additional bipartite concept matching to ensure an injective mapping between offer and request concepts. Details on the filters can be found in (Kapahnke et al, 2008).

2.2. Syntactic Operation Matching

SAWSDL-MX implements the same similarity measures as that of OWLS-MX, which are the Loss-of-Information, the Extended Jaccard, the Cosine and the Jensen-Shannon similarity measures. Also the architecture of SAWSDL-MX allows the integration of other text similarity measures such as those provided by SimPack which is also used in the iMatcher matchmaker (Kiefer et al, 2008). The weighted keyword vectors of inputs and outputs for every operation are generated by first unfolding the referenced concepts in the ontologies. The resulting set of primitive concepts of all input concepts of a service operation is then processed to a weighted keyword vector based on TFIDF (Term Frequency and Inverse Document Frequency) weighting scheme, the same is done with its output concepts. The text similarity of a service offer operation and a request operation is the average of the similarity values of their input and output vectors according to the selected text similarity measure.

2.3 Hybrid Operation Matching

SAWSDL-MX combines logic-based and syntactic-based matching to perform hybrid semantic service matching. There are different options of combination:

- A compensative variant uses syntactic similarity measures in situation where the logic-based filters don't apply with respect to logic-based false negatives. It helps to improve the service ranking by re-considering them again in the light of their computed syntactic similarity.
- An integrative variant deals with problems concerning logic-based false positives by not taking the syntactic similarity of concepts into account only when a logical matching fails, but as a conjunctive constraint in each logical matching filter.

SAWSDL-MX inherited the compensative variant from OWLS-MX.

2.4. Limitations of SAWSDL-MX

We identified the following limitations amongst others:

- Users are not afforded flexibility to specify queries for services of interest to them using descriptions that are expressed using terms in their own ontologies i.e. ontology heterogeneity problem still persists.
- It lacks support for service selection based on non functional specification of offered or registered services

To address these limitations, we propose solutions that serve as extension and improvement on SAWSDL-MX. These extensions are:

1. We added a component to SAWSDL-MX that allows for specifying mappings from a user ontology to relevant domain ontologies used to specify Web Services. This is to take care of problem of ontology heterogeneity.
2. We also describe how user-specified preferences for Web Services in terms of non-functional requirements (e.g., QoS) can be incorporated into the Web Services discovery mechanism to generate a partially ordered list of services that meet user-specified functional requirements.

We give details of our proposal in the next section

3. Proposed Framework

This section describes our framework for ontology-based flexible discovery of Semantic Web Services which is an extension proposed for SAWSDL-MX.

3.1 Mapping User Ontology to Domain Ontology

Ontologies are the basis for shared conceptualization of a domain, and comprise of concepts with their relationships and properties (Gruber, 1993). In the work of Caragea et al (2004), a precise definition of ontology was given which we adopt in this work. They looked at ontology in terms of hierarchy, that if we define S to be a partially ordered set under ordering \leq . Then an ordering \preceq defines a hierarchy for S if the following three conditions are satisfied:

- (1) $x \preceq y \rightarrow x \leq y; \forall x, y \in S$. Then (S, \preceq) is better than (S, \leq) ,
- (2) (S, \preceq) is the reflexive, transitive closure of (S, \preceq)
- (3) No other ordering exists that satisfies (1) and (2).

Then all ontology does is to associate orderings to their corresponding hierarchies. This means that given a set of concepts that define a domain, ontology associates ordering among the concepts i.e. they can be parsed as hierarchical tree with the nodes denoting the different concepts and the edges, relationship between these concepts. In order to make ontologies interoperable, so that the terms in different

ontologies are brought into correspondence, we need to define mappings. These mappings are specified through

Interoperation Constraints. Which they define as:

Let (H_1, \preceq_1) and (H_2, \preceq_2) , be any two hierarchies. Then the set of Interoperation Constraints (IC) the set of relationships that exist between elements from two different hierarchies. For two elements, $x \in H_1$ and $y \in H_2$, we can have one of the following Interoperation Constraints:-
 $x : H_1 = y : H_2, x : H_1 \neq y : H_2, x : H_1 \leq y : H_2,$
and,

$$x : H_1 \preceq y : H_2.$$

This means that given two ontologies defined as above, they interoperate if we can define a relation between the elements (which are actually concepts in the respective domains). Equality could mean they occur at the same level in the tree structure. Also, less than could mean a parent-child relation and so on.

The interoperation constraint gives a fine mapping from user ontology to the domain ontology used by the matchmaker.

3.2 Incorporating Non-Functional Requirements

Quality of Service can be defined as a set of non-functional attributes that may have significant effect on the service quality offered by a Web Services. Examples of these non-functional attributes include scalability, performance, availability etc. Different QoS attributes might be important in different applications and different classes of Web Services might use different sets of non-functional attributes to specify their QoS properties. For example, response time may be an important QoS criterion for a service which provides online voice conferencing, as opposed to, availability for a service which provides online reservation. As a result, we categorize them into:

Domain Independent Attributes: The domain-independent attributes represent those QoS characteristics which are not specific to any particular service (or a community of services). Examples include scalability, throughput etc.

Domain Dependent Attributes: domain-dependent attributes capture those QoS properties which are specific to a particular domain. For example in a gift packaging domain, gift decoration rating may be a QoS attribute.

This gives rise to a situation where a user might consider some non-functional attributes valuable for his/her purpose (and hence, defined in the user ontology). These attributes are used to compose a quality vector comprising of their values for each **candidate service** (services that satisfy functional requirement of the user). These quality vectors are used to derive a quality matrix, Q .

Doina et al, (2005) defined a quality matrix, $Q = \{V(Q_{ij}); 1 \leq i \leq m; 1 \leq j \leq n\}$, which refers to a collection of quality of service attribute-values for a set of candidate services, such that, each row of the matrix

corresponds to the value of a particular QoS attribute (in which the user is interested) and each column refers to a particular candidate service. In other words, $V(Q_{ij})$, represents the value of the i^{th} QoS attribute for the j^{th} candidate service. These values are obtained from the profile of the candidate service providers and mapped to a scale between 0 & 1 by applying standard mathematical maximization and minimization formulas based on whether the attribute is positive or negative. For example, the values for the attributes Latency and Fault Rate needs to be minimized, whereas Availability needs to be maximized. Also, to give relative importance to the various attributes, the users can specify a weight value for each attribute, which are used along with the QoS attribute values to give relative scores to each candidate service using an additive value function, f_{QoS} . Formally,

$$f_{QoS}(service_j) = \sum_{i=1}^m (V(Q_{ij}) \times weight_i)$$

where, m is the number of QoS attributes in \mathcal{Q} (Doina et al, 2005).

This incorporates non-functional requirements into the discovery process

4. Proposed Architecture

This section shows how the solutions identified with SAWSDL-MX1.0 are incorporated to have an improved and refined prototype. Basic aspects of SAWSDL-MX1.0 are maintained. The figure 1 gives the architecture.

The proposed new SAWSDL-MX consists of all the components of SAWSDL-MX1.0 as found in Kapahnke et al (2008). The new components include QoS matcher and Mapping User to Domain Ontology which in the next section. From the perspective of service providers, the new SAWSDL-MX allows the registration of SAWSDL Web Services offers along with their QoS ratings, at the service registry. For requesters, SAWSDL-MX provides an interface for submitting queries by means of a SAWSDL document specifying details about the desired service interface in the user's ontology. After which the user's ontology is mapped to the domain ontology of the matchmaker, as is in SAWSDL-MX, the domain ontology in our proposed framework is also OWL-based. After the service discovery process using the logic, syntactic or hybrid filters, QoS matcher of the proposed improved SAWSDL-MX takes as input those services returned by the filters and calculate their additive value function (f_{QoS}) scores and returns a ranked list of service offers with f_{QoS} scores greater than user specified threshold. This guarantees that the returned services match both functional and non-functional specification of the query

4.1 Mapping User to Domain Ontology:

This component takes the user ontology and maps it to the domain ontology of the matchmaker. Ontologies associate

orderings to their corresponding hierarchies. For example, let $S = \{Food, ChineseFood, SeaFood\}$ (Figure 2). We can define the partial ordering \leq on S according to an *is-a* (or sub-class) relationship. For example, *SeaFood is-a* sub-class of *ChineseFood*, *ChineseFood is-a* sub-class of *Food* and, also *SeaFood is-a* sub-class of *Food*. Besides, every class can be regarded as a sub-class of itself. Thus, $(S, \leq) =$

$\{(ChineseFood, ChineseFood), (SeaFood, SeaFood), (Food, Food), (SeaFood, ChineseFood), (SeaFood, Food), (ChineseFood, Food)\}$

, is the reflexive, transitive closure of the set:

$(S, <) = \{(ChineseFood, Food), (SeaFood, ChineseFood)\}$

, which is the only hierarchy associated with (S, \leq) . We

defined interoperability constraints to specify mapping from user to domain ontology. For example, let $O_{Chinese}^U$ be user

ontology for specifying Chinese food different from $O_{ChineseFood}$ (domain ontology), assuming that the

ontologies $O_{Chinese}^U$ and $O_{ChineseFood}$ associate *is-a*

orderings to their corresponding hierarchies, we can have the following interoperation constraints, among others-

$Chicken : H_{Chinese}^U = Poultry : H_{ChineseFood}$

$Fish : H_{Chinese}^U = SeaFood : H_{ChineseFood}$

$Chicken : H_{Chinese}^U \neq Appetizer : H_{ChineseFood}$,and so

on.

4.2 QoS Matcher

This component incorporates non-functional aspects into the discovery process. Definition (**Candidate Service Providers**): Let $\mathcal{S} = \{S_1, \dots, S_n\}$ denote the set of

services which are available (or registered with our system). We call, $\mathcal{S}' \subseteq \mathcal{S}$, the set of candidate providers, if they meet

the requested functional properties of the user in terms of inputs, output, precondition and effects (IOPE's). This

component takes as input, the candidate services returned by either logic-based or IR-based or hybrid matcher and return

services that satisfy functional specification and have overall score (for the non-functional attributes) greater than

some threshold value specified by the user. If several services satisfy these constraints, it is at the discretion of the

user. But, if no service exists, then an exception is raised and the user is notified appropriately.

The Service Requester specifies a request for service using the Service Requesting API. Such a request is described

using OWL-DL. That allows to apply standard subsumption reasoning used for OWL-S MX. The requester also specifies

the interoperation constraints (ICs) between the terms and concepts of its ontologies to the domain ontologies.

For our first prototype, the constraints are defined manually. With the help of these translations, the service requesting

API transforms the requester's query, into a domain-specific query. The matchmaking engine then tries to find service

advertisement(s) which match the user's request. This process is the same as SAWSDL-MX1.0 and it returns a set

of candidate service providers which is taken as input by the QoS Matcher. For a particular service request query, our system selects one or more services which satisfies user's constraints (in terms of functional requirement) and has an overall score (for the non-functional attributes) greater than some threshold value specified by the user. But, if no service exists, then an exception is raised and the user is notified appropriately. The QoS Algorithm is given in 4.3

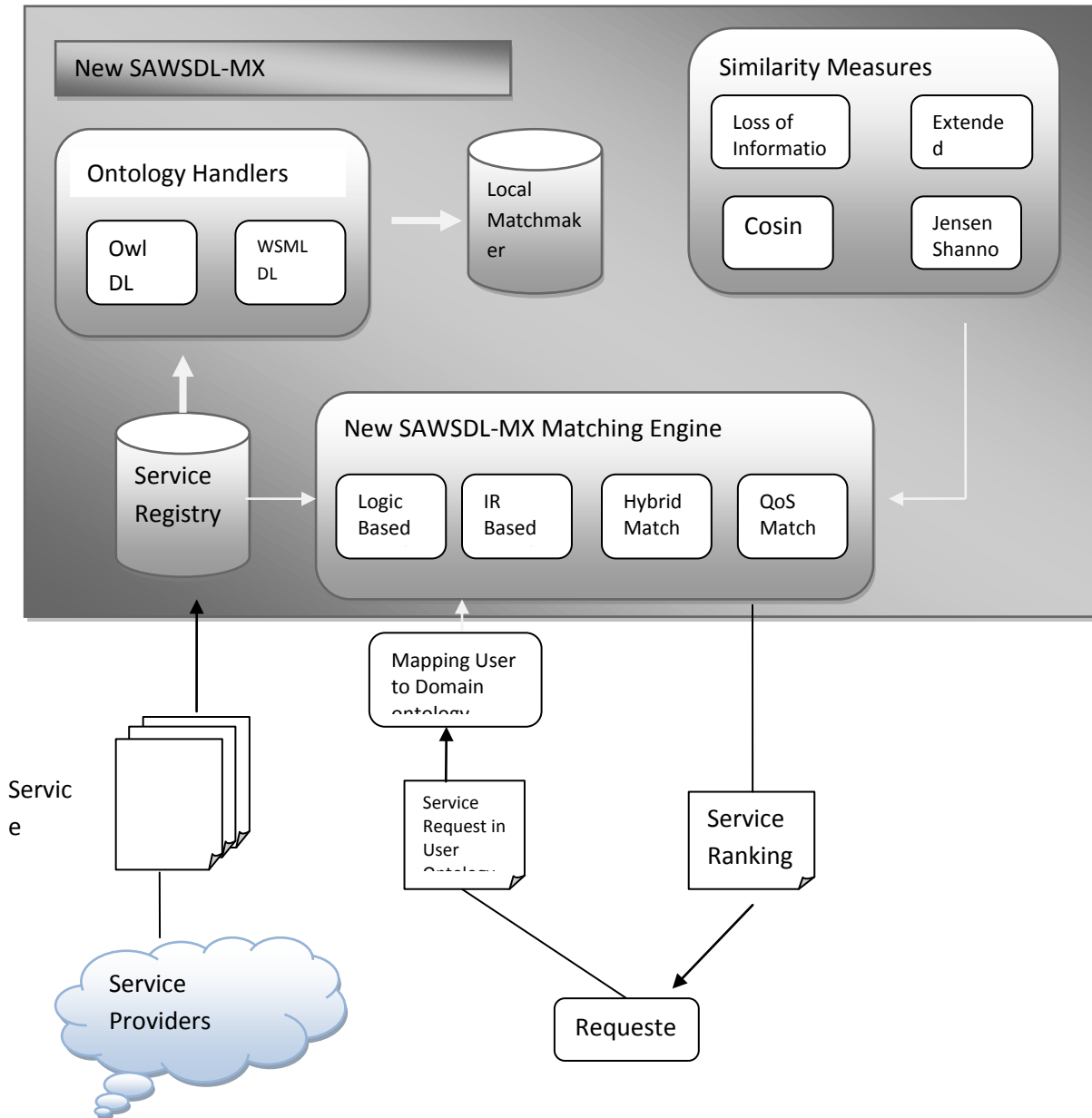


Figure 1: New SAWSDL-MX

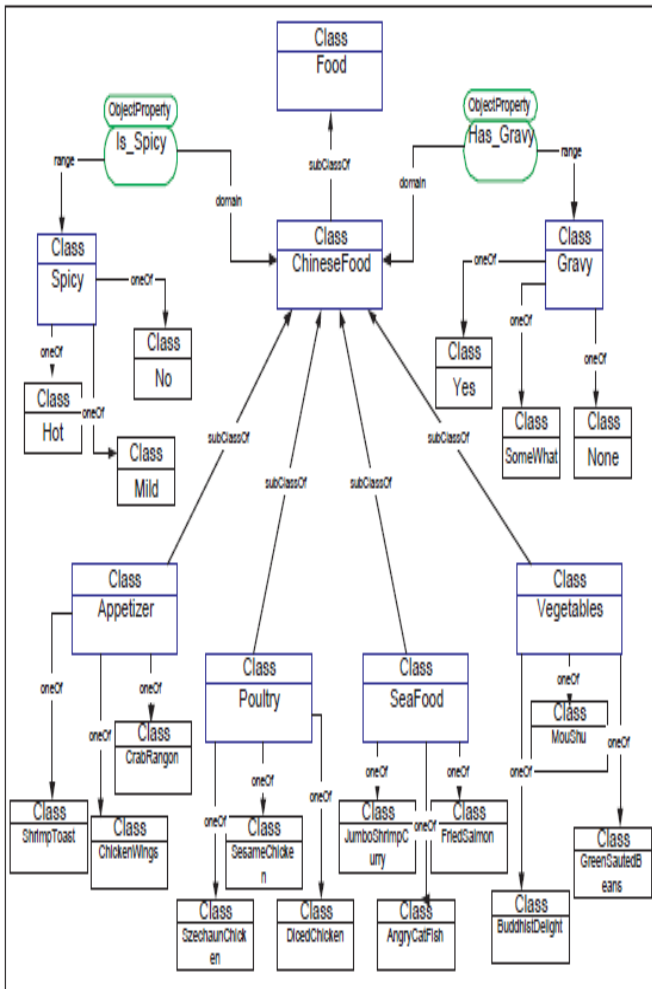


Figure 2: Domain Ontology for Chinese Food

4.3. QoS Algorithm

Finds services which belong to candidate services that best match a requesters non-functional requirements A_i ; it returns set of services $(Service_j, f_{QoS})$ with additive value function

$$f_{QoS} \geq \text{threshold score vale } (\varphi)$$

Function $QoS(Attribute A_i, Weight_i, \varphi)$

local $Q, result$

$$Q = \{V(Q_{ij}); 1 \leq i \leq m; 1 \leq j \leq n\}$$

for all Candidate_{Service} do

$$f_{QoS}(Service_j) = \sum_{i=1}^m (V(Q_{ij}) \times Weight_i)$$

if $f_{QoS} \geq \varphi$ then

result := result \cup $\{(Service_j, f_{QoS})\}$

end if

end for

return result

end Function

5. Service Discovery using Proposed Framework

We now give a detailed example to ascertain the workability of our framework.

Example 1: Let $Z = \{S_1, S_2, S_3, S_4\}$ be the set of services returned that satisfy the functional requirement of the user based on logic, syntactic or hybrid filters. Now if the user is interested in scalability, availability, ratings, latency and throughput. Assuming based on the QoS attributes specified in the service description, the quality matrix is given below

$$Q = \begin{pmatrix} \text{Attributes / Service} & S_1 & S_2 & S_3 & S_4 \\ \text{scalability} & 0.85 & 0.73 & 0.67 & 0.3 \\ \text{Availability} & 0.85 & 0.84 & 0.09 & 0.25 \\ \text{Rating} & 0.91 & 0.94 & 0.5 & 0.35 \\ \text{Latency} & 0.05 & 0.07 & 0.11 & 0.01 \\ \text{Throughput} & 0.83 & 0.96 & 0.55 & 0.23 \end{pmatrix}$$

The additive value function for each service is given below

$$\begin{pmatrix} \text{Service} & S_1 & S_2 & S_3 & S_4 \\ f_{QoS} & 2.02 & 1.96 & 1.0 & 0.67 \end{pmatrix}$$

Assuming the user specifies

$weight(scalability) = 0.8, weight(availability) = 0.9,$

$weight(rating) = 0.5, weight(latency) = 0.7,$

$weight(throughput) = 0.1$ and $threshold score(\varphi) = 1$

Clearly, we see that only services S_1, S_2 and S_3 will be returned after calculating their respective f_{QoS} scores. There will be implicit ranking of these returned services based on the one with the highest f_{QoS} score to the lowest.

6. Future

In this paper, we presented only the framework of our proposed matchmaker, the implementation is however, an ongoing work. Also in this initial framework, interoperability constraints are specified manually by the user, a possible future work is to incorporate (semi) automatic correspondences between user and domain ontology should be considered. Also, allowing the matchmaker to accommodate model reference to multiple ontologies description will further enhance its capabilities.

7. Related Work

Ontologies have been identified as the basis for semantic annotation that can be used for discovery. Ontologies are the basis for shared conceptualization of a domain (Gruber, 1993), and comprise of concepts with their relationships and properties. Use of ontologies to provide underpinning for information sharing and semantic interoperability has been long realized (Gruber et al, 1991; Kashyap et al, 1994;

Wache et al, 2001). By mapping concepts in a Web resource (whether data or Web Services) to ontological concepts, users can explicitly define the semantics of that resource in that domain. The semantic relationships in ontologies are machine readable, and thus enable inferencing and asking queries about a subject domain. The W3C standard OWL (www.w3.org/TR/2004/REC-owl-guide-20040210) is based on RDF(S) and supports the modeling of knowledge/ontologies (Herrmann et al, 2006). OWL supports developing of ontologies in a powerful way, but lacks in describing the technical details of services. WSDL-S (adopted in industry as SAWSDL) extends WSDL in order to use semantic capabilities of OWL to provide semantically enriched meanings of WSDL services – WSDL-S connects WSDL and OWL in a practical way (Akkiraju et al, 2005). In the work of Pierre (2007), a WSDL-S to UDDI mapping was defined and a service publication and querying API named LUCAS (layer for UDDI compatibility with annotated semantics) is placed around it. Jyotishman et al (2005) proposes a framework for ontology-based flexible discovery of Semantic Web Services. The proposed approach relies on user-supplied, context-specific mappings from a user ontology to relevant domain ontologies used to specify Web Services. A description of how user-specified preferences for Web Services in terms of non-functional requirements (e.g., QoS) can be incorporated into the Web Services discovery mechanism to generate a partially ordered list of services that meet user-specified functional requirements. Our approach is similar to this in the sense that both provide mappings from user to domain ontology however, their approach is OWL based while ours is SAWSDL based. OWL-S/UDDI (Srinivasan et al, 2004)) matchmaker combines UDDI's proliferation into the Web Services infrastructure and OWL-S's explicit semantic description of the Web Services. Glue (www.swa.cefriel.it/Glue) is a WSMO compliant discovery engine that aims at developing an efficient system for the management of semantically described Web Services and their discovery. OWL-S MX by Matthias et al (2008) is a hybrid OWL-S semantic service matchmaking algorithm. It uses both logical based and content based retrieval techniques for Web Services discovery. The hybrid semantic service matching uses six different filters to calculate the degree of semantic match between the request and advertisement. The first four filters are purely logic based and the next two are hybrid, which use the IR similarity metric values. SAWSDL –MX by Kapahnke et al (2008) is an approach that does hybrid semantic Web Services discovery using SAWSDL based on logic based matching as well as information retrieval based techniques. It is significantly based on but a further refinement of OWL-MX. Our work extended the work of Kapahnke et al. as stated earlier.

8. Conclusion

It is a fact that users will always want to request for services of interest to using terms defined in their own ontology. So,

we included a mechanism that allows mapping from user ontology to the domain ontology used to describe the services. This mapping is specified in form of interoperability constraints between the user and domain ontologies. Since ontology associates ordering to corresponding hierarchy, if we can associate a concept from user ontology to another in the domain ontology, we can then form a relation between all the concepts in the respective domain i.e. user and domain ontology. Also, Web Services are distributed as well as autonomous by their very nature, and can be invoked dynamically by third parties over the Internet, their QoS can vary greatly. Thus, it is vital to have an infrastructure which takes into account the QoS provided by the service provider and the QoS desired by the service requester, and ultimately find the (best possible) match between the two during service discovery. Integrating QoS features in the profile of Web Services is to the advantage of both users and providers. QoS profiles of Web Services are crucial in determining which service best addresses the user desires and objectives. If the discovered Web Services are accompanied with descriptions of their non-functional properties, then the automated Web Service selection and composition that takes place, considers the user's QoS preferences in order to optimize the user's Web Service-experience regarding features such as performance, reliability, security, integrity, and cost. On the other hand, QoS can give Web Service providers a significant competitive advantage in the e-business domain, as QoS-aware services meet user needs better and thus attract more customers. As major contribution, the extension of SAWSDL-MX to allow for mapping of user to domain ontology, gives users the flexibility to make queries without having to use the domain ontology. Also the incorporation of non-functional specification in the discovery process of SAWSDL-MX makes it user centered in that it returns what the user wants bearing in mind tradeoffs that may exist.

8.1 Limitations of our Framework

Potential of the proposed Framework is illustrated in the examples in the earlier section. However, a more concrete feel can be obtained when the proposed Framework is implemented. Also, our proposed framework takes care of only problem of ontology heterogeneity. Problem of ontology description heterogeneity still persists.

9. Acknowledgements

I will first of all like to register my profound gratitude to God Almighty for the opportunity to start and successfully complete this program. My sincere gratitude goes to my

wonderful and accomplished supervisor Prof. S. B. Junaidu, for the guidance and free will to express myself in the course of this research. Sir, I am indeed grateful. Also, I will like to say a big thank you to my minor supervisor in the person of Dr. N. Choji. To my lecturers, I say a bucketful of thanks and pray God grants each of your heart desires. My colleagues in the department, what can I say but a gracious thanks for your support and encouragements. Ha, Mr. Kana, what can I have done without your encouragements and support? May God grant your heart desires according to his riches in glory. This acknowledgement can never be complete without saying big up to you my wonderful parents, brothers, sisters, nieces, nephews and all relatives. You all are dear to me as such I register my warmest appreciation and say “thank you all”

10. References

1. Akkiraju, R. J., Farrell, J., Miller, A., Nagarajan, M., Sheth A. and Verma K. (2005). Web Services Semantics – WSDL-S [online] Available <http://www.w3.org/2005/04/FSWS/Submissions/17/WSDL-S.htm>.
2. Ankolenkar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D.L., McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne T.R., and Sycara, K. (2002). The DAML Services Coalition, "DAML-S: Web Services Description for the Semantic Web", The First International Semantic Web Conference (ISWC), Sardinia (Italy).
3. Business Process Execution Language for Web Services Version 1.1. [online] available <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>
4. Caragea, D., Pathak, J. and Honavar V. (2004). Learning Classifiers from Semantically Heterogeneous Data Sources. In 3rd Intl. Conference on Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems.
5. Cardoso, J. and Sheth A. (2002). Semantic e-Workflow Composition. Journal of Intelligent Information Systems (JIIS), 21(3), 191-225. Kluwer Academic Publishers
6. Cardoso, J., Miller, J. and Emani, S. (2008). Web Services Discovery Using Annotated WSDL. In: Reasoning web Fourth International Summer School 2008 Springer.
7. Curbera, F., Nagy, W. and Weerawarana, S. (2001). Web Services: Why and How. In Workshop on Object-Oriented Web Services – OOPSLA 2001, Tampa, Florida, USA.
8. DAML-S 0.7 Draft Release, 2002.
9. Duftler, M.J., Mukhi, N.K., Slominski, A. and Weerawarana, S. (2001). Web Services Invocation Framework.
10. Duy Ngan L., Soong Goh, A. E., Tru Hoang C. (2007). A Survey of Web Services Discovery Systems: International Journal of Information Technology and Web Engineering, Vol. 2, Issue 2.
11. Fensel, D., and Bussler, C. (2002). The Web Services Modeling Framework WSMF. In: Electronic Commerce Research and Applications, Vol. 1, Issue 2, Elsevier Science B.V.
12. Garofalakis, J., Panagis, Y., Sakkopoulos, E., and Tsakalidis, A. (2004). Web Services Discovery Mechanisms: Looking for a Needle in a Haystack? Paper presented at the International Workshop on Web Engineering
13. GLUE. Available from: <http://swa.cefril.it/Glue>
14. Gruber, T.R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. In J. A. Allen, R. Fikes, and E. Sandewall, (Ed), San Mateo, CA., Morgan Kaufman
15. Gruber, T.R. (1993). A Translation Approach to Portable Ontology Specifications.” Knowledge Acquisition, 5(2), 199-220.
16. Herrmann, M., Golden, R. (2006). Why SOA lacks reuse. In: JBoss World 2006, http://jbossworld.com/jbvw 2006/soa_for_the_real_world/HERRMANN_CO-Layer_final.pdf
17. Herrmann, M., Muhammad, A. A. and Dalferth, O. (2007). Applying Semantics (WSDL, WSDL-S, OWL) in Service Oriented Architectures (SOA): 10th Intl. Protégé Conference.
18. <http://twwww.w3.org/TR/sawSDL/>
19. Jyotishman P., Neeraj, K., Doina, C., Vasant, G. H. (2005). A Framework for Semantic Web Services Discovery. In ACM 7th Intl. workshop on Web Information.
20. Kapahnke, P. and Mathias, K. (2008). Semantic Web Services Selection with SAWSDL-MX.
21. Kashyap, V. and Sheth, A. (1994). Semantics-based Information Brokering. In Proceedings of the Third International Conference on Information and Knowledge Management (CIKM).
22. Kaufer, F. and Klusch, M. (2006). WSMO-MX: A Logic Programming Based Hybrid Service Matchmaker. Proceedings of the 4th IEEE European Conference on Web Services (ECOWS 2006), IEEE CS Press, Zurich, Switzerland.

23. Keller, U. (2004). WSMO Web Services Discovery.
24. Kiefer, C. and Bernstein, A. (2008). The Creation and Evaluation of iSPARQL Strategies for Matchmaking. Proceedings of the 5th European SemanticWeb Conference (ESWC), Lecture Notes in Computer Science, Vol. 5021, pages 463–477, Springer-Verlag Berlin Heidelberg.
25. Matthias, K., Benedikt, F. and Mahboob K. (2008). OWLS-MX: A hybrid SemanticWeb Services matchmaker for OWL-S services. Web Semantics: Science, Services and Agents on the World Wide Web.
26. Menascé, D. A. (2002). QoS Issues in Web Services. IEEE Internet Computing, vol. 6, no. 6, pp. 72-75.
27. Miller J., Verma K., Rajasekaran P., Sheth A., Aggrawal R., and Sivashanmugam K. (2004). WSDL-S: A Proposal to W3C WSDL 2.0 Committee, <http://lsdis.cs.uga.edu/projects/WSDL-S/wSDL-s.pdf>
28. OWL (Ontology Web Language); W3C Recommendation <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>
29. Paolucci, M., Kawamura, T., Payne, T.R. and Sycara, K. (2002). Importing the Semantic Web in UDDI. Appeared In Web Services, E-Business and Semantic Web Workshop.
30. Paolucci, M., Kawamura, T., Payne, T.R. and Sycara, K. (2002). Semantic Matching of Web Services Capabilities. Proceedings of the 1st International Semantic Web Conference.
31. Patil, A., Oundhakar, S. and Sheth, A. (2003). Semantic Annotation of Web Services, Technical Report, LSDIS Lab, Department of Computer Science, University of Georgia.
32. Pierre, C. (2007). Toward a Semantic Web Services discovery and dynamic orchestration based on the formal specification of functional domain knowledge. Thales Land & Joint Systems, LIP6 Computer Science Laboratory in proceedings of ICSSEA.
33. Radatz, J. and Sloman, M. S. (1988). A Standard Dictionary for Computer Terminology: Project 610. IEEE Computer, 21(2).
34. Simple Object Access Protocol (SOAP) 1.1, <http://www.w3.org/TR/soap/>.
35. SOAP Version 1.2 Part 1: Messaging Framework [online] Available <http://www.w3.org/TR/soap12-part1/>. Srinivasan, N., Paolucci, M. and Sycara, K. (2004). Adding OWL-S to UDDI, implementation and throughput. Paper presented at the First International Workshop on Semantic Web Services and Web Process Composition. Dan Diego, California, USA.
36. Sycara, K., Paolucci, M., Ankolekar, A. and Srinivasan, N. (2003). Automated discovery, interaction and composition of Semantic Web Services. Journal of Web Semantics, vol 1, Elsevier.
37. Tversky, A. (1977). Features of Similarity. Psychological Review. 84(4): p. 327-352.
38. UDDI Version 3.0 [online] <http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>
39. Universal Description, Discovery and Integration (UDDI), <http://www.uddi.org/>
40. Verma, K. (2005). A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services. Journal of Information Technology and Management.
41. Verma, K. (2007). Allowing the use of Multiple Ontologies for Discovery of Web Services in Federated Registry Environment: Athens. p. 1-27.
42. Wache, H., V ogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hubner, S. (2001). Ontology-based integration of information - a survey of existing approaches. In: Stuckenschmidt, H., (Ed.), IJCAI-01 Workshop: Ontologies and Information Sharing, 108-117.
43. Walsh, A. E. (2002). UDDI, SOAP, and WSDL: The Web Services Specification Reference Book (1st edition ed.) 0130857262: Pearson Education
44. Web Ontology Language for Web Services, <http://www.daml.org/services/>
45. Web Services Description Language (WSDL) [online] Available <http://www.w3.org/TR/wsdl>.
46. Web Services Modeling Ontology, <http://www.wsmo.org/>
47. WSDL: Web Services Description Language, <http://www.w3.org/TR/wsdl>.
48. WSML. Web Services Modeling Language (WSML). 2004 [cited 2004; Available from:<http://www.wsmo.org/wsml/index.html>.

A Novel Approach towards Cost Effective Region-Based Group Key Agreement Protocol for Ad Hoc Networks using Chinese Remainder Theorem

K. Kumar¹, J.Nafeesa Begum² and Dr.V. Sumathy³

1. Research Scholar & Lecturer in CSE, Government College of Engg, Bargur- 635104, Tamil Nadu, India
pkk_kumar@yahoo.com

2. Research Scholar & Sr. Lecturer in CSE, Government College of Engg, Bargur- 635104, Tamil Nadu, India

nafeesa_jeddy@yahoo.com

3. Asst. Professor in ECE, Government College of Technology, Coimbatore, Tamil Nadu, India

sumi_gct2001@yahoo.co.in

Abstract - A group key agreement (GKA) protocol is a mechanism to establish a cryptographic key for a group of participants, based on each one's contribution, over a public network. In ad-hoc networks, the movement of the nodes may quickly change the topology resulting in the increased overhead during messaging for topology maintenance. The Region-based schemes of ad-hoc networks, aim at handling topology maintenance, managing node movement and reducing overhead. In this paper, a simple, secure and efficient Region-based GKA protocol using CRTDH&TGDH well suited to dynamic ad-hoc networks is presented and also introduces a region-based contributory group key agreement that achieves the performance lower bound by utilizing a novel CRTDH and TGDH protocol called CRTDH&TGDH protocol. Both theoretical and experimental results show that the proposed scheme achieves communication, computation and memory cost is lower than the existing group key agreement schemes.

Keywords: Chinese Remainder theorem, Ad hoc networks, Region-Based key agreement

1. Introduction

Ad hoc networks are networks composed of constrained devices communicating over wireless channels in the absence of any fixed infrastructure. Moreover, network composition is highly dynamic when devices leave /join the network quite frequently. Securing this type of networks become a more difficult task with additional challenges in the form of lack of trusted third parties, expensive communication, ease of interception of messages and limited computational capabilities of the devices. In ad hoc networks, the key distribution techniques are not useful as there is not enough trust in the network so as to agree on a key decided by one member or some central authority.

Group Key Agreement (GKA) protocols [4,5, 6, 7, &8], which enable the participants to agree on a common secret value based on each participant's contribution, seem to provide a good solution. Also when group composition changes, group controller can employ supplementary key agreement protocols to get a new group key.

Secure group communication (SGC) is the process by which members in the group can securely communicate with each other and information being shared is inaccessible to anybody outside the group. A SGC protocol should efficiently manage the group key when the members join and leave the groups. This is considerably done in MANETs with its high mobility and dynamic network topology.

SGC should satisfy the following properties: shared group key, backward secrecy, forward secrecy, multiple users to join/leave simultaneously, efficiency with minimum amount of computation and communication.

A number of protocols have been proposed to handle SGC over MANETs and none of the above properties are satisfied. In our paper, we have proposed the important MANET features with respect to SGC: No pre-shared secret since the participating members are not known before hand, No centralized Trusted Authority (TA), Optimized battery power, all nodes have balanced load, position and capability and Nodes are highly mobile.

Earlier we have proposed [1] a Contributory Group Key Agreement which fulfills the efficacious lower bound by utilizing a novel GECDH & TGECDH in which a leader communicates with the member in the same region using a regional key KR. The Outer Group Key is derived from subgroup leaders. In this approach, a GECDH protocol needs member serialization among the members. It is not a desirable property of ad hoc networks.

We propose a communication and computation efficient group key agreement protocol in ad-hoc network. In large and high mobility ad hoc networks, it is not possible to use a single group key for the entire network because of the enormous cost of computation and communication in rekeying. So, we divide the group into several subgroups; let each subgroup has its subgroup key shared by all members of the subgroup. Each group has sub group controller node and gateway node, in which the sub group controller is controller of each subgroup and gateway node is controller of among subgroups. Let each gateway member contribute a partial key to agree with a common Outer group key among the subgroups.

In this paper we have proposed a Region-based Contributory GKA that achieves the performance lower bound by utilizing a novel Chinese Remainder Theorem Diffie-Hellman (CRTDH) and TGDH protocol for secure group communication over MANETs. In this approach, the member serialization between the subgroup members is eliminated.

The contribution of this work includes:

1. In this paper, we propose a new efficient method for solving the group key management problem in ad-hoc network. This protocol provides efficient, scalable and reliable key agreement service and is well adaptive to the mobile environment of ad-hoc network.
2. We introduce the idea of subgroup and subgroup key and we uniquely link all the subgroups into a tree structure to form an outer group and outer group key. This design eliminates the centralized key server. Instead, all hosts work in a peer-to-peer fashion to agree on a group key. We use Region-Based Group Key Agreement (RBGKA) as the name of our protocol. Here we propose a region based group key agreement protocol for ad hoc networks using Chinese Remainder Theorem called Region-Based CRTDH & TCDH protocol.
3. We design and implement Region-Based Group key agreement protocol using Java and conduct extensive experiments and theoretical analysis to evaluate the performance like memory cost, communication cost and computation cost of our protocol for Ad- Hoc network.

The rest of the paper is, Section 2 describes the Chinese Remainder Theorem. Section 3 presents the proposed CRTDH&TGDH schemes. Section 4 describes CRTDH Key Agreement Protocol. Section 5 describes the TGDH Protocol. Section 6 describes the experimental results. Section 7 describes the complexity analysis and finally Section 8 concludes the paper.

2. Chinese Remainder Theorem(CRT)

The Chinese Remainder Theorem is used as the theoretical foundation in many cryptosystems.

Suppose N_1, \dots, N_r are pairwise relatively prime positive integer, i.e., $\gcd(N_i, N_j) = 1$ if $i \neq j$, and let $N = N_1 \cdot N_2 \cdot \dots \cdot N_r$. For any given integers a_1, \dots, a_r , consider the following system of congruences:

$$\begin{aligned} X &\equiv a_1 \pmod{N_1} \\ X &\equiv a_2 \pmod{N_2} \\ &\vdots \\ X &\equiv a_r \pmod{N_r} \end{aligned}$$

Then the system has a unique solution modulo N , namely:

$$x \equiv \sum_{i=1}^r a_i n_i y_i \pmod{N}$$

Where $n_i = N/N_i$ and $y_i = n_i^{-1} \pmod{N_i}$ (i.e. y_i is the multiplicative inverse of n_i modulo N_i).

3. Proposed Scheme

3.1. Motivation

There has been a growing demand in the past few years for security in collaborative environments deployed for emergency services where our approach can be carried out very efficiently are shown in Fig.1. Confidentiality becomes one of the top concerns to protect group communication data against passive and active adversaries. To satisfy this requirement, a common and efficient solution is to deploy a group key shared by all group application participants. Whenever a member leaves or joins the group, or whenever a node failure or restoration occurs, the group key should be updated to provide forward and backward secrecy.



Figure.1. Secure Group Applications

Therefore, a key management protocol that computes the group key and forwards the rekeying messages to all legitimate group members is central to the security of the group application.

In many secure group applications [1, 3], a Region based contributory GKA schemes may be required. In such cases, the group key management should be both efficient and fault-tolerant. In this paper,



Figure.2. Battlefield Scenario

we describe a military scenario (Figure.2). A collection of wireless mobile devices are carried by soldiers or a Battlefield tanks. These mobile devices cooperate in relaying packets to dynamically establish routes among themselves to form their own network “on the fly”. However, all nodes except the one with the tank, have limited battery power and processing capacities. For the sake of power- consumption and computational efficiency, the tank can work as the Gateway member while a contributed group key management scheme is deployed.

3.2. System Model

3.2.1. Overview of Region-Based Group Key Agreement Protocol:

The goal of this paper is to propose a communication and computation efficient group key establishment protocol in ad-hoc network. The idea is to divide the multicast group into several subgroups, let each subgroup has its subgroup key shared by all members of the subgroup. Each Subgroup has subgroup controller node and a Gateway node, in which Subgroup controller is controller of subgroup and a Gateway node is controller of subgroups controller.

For example, in Figure.3, all member nodes are divided into number of subgroups and all subgroups are linked in a tree structure as shown in Figure.4.

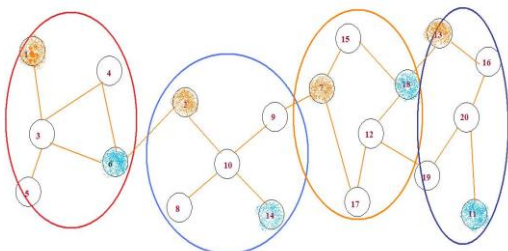


Figure.3: Members of group are divided into subgroups

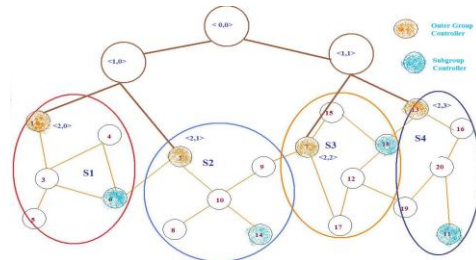


Figure.4: Subgroups link in a Tree Structure

The layout of the network is as shown in below figure.5.

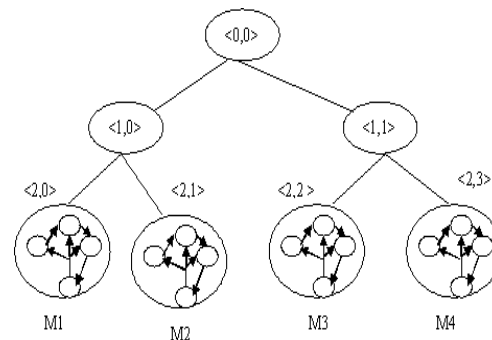


Figure.5. Region based Group Key Agreement

One of the members in the subgroup is subgroup controller. The last member joining the group acts as a subgroup controller. Each outer group is headed by the outer group controller. In each group, the member with high processing power, memory, and Battery power acts as a gateway member. Outer Group messages are broadcast through the outer group and secured by the outer group key while subgroup messages are broadcast within the subgroup and secured by subgroup key.

Let N be the total number of group members, and M be the number of the subgroups in each subgroup, then there will be N/M subgroups, assuming that each subgroup has the same number of members.

There are two shared keys in the Region-Based Group Key Agreement Scheme:

1. Outer Group Key (KG) is used to encrypt and decrypt the messages broadcast among the subgroup controllers.
2. The Subgroup Key (KR) is used to encrypt and decrypt the Sub Group level messages broadcast to all sub group members.

In our Region-Based Key Agreement protocol shown in Fig.5 a Subgroup Controller communicates with the member in the same region using a Regional key (i.e Sub group key) KR. The Outer Group key KG is derived from the Outer Group Controller. The Outer Group Key KG is used for secure data communication

among subgroup members. These two keys are rekeyed for secure group communications depending on events that occur in the system.

Assume that there are total N members in Secure Group Communication. After sub grouping process (Algorithm 1), there are S subgroups $M_1, M_2 \dots M_s$ with $n_1, n_2 \dots n_s$ members.

Algorithm. 1. Region-Based Key Agreement protocol

1. The Subgroup Formation

The number of members in each subgroup is $N / S < 100$.

Where, N – is the group size. And S – is the number of subgroups.

Assuming that each subgroup has the same number of members.

2. The Contributory Key Agreement protocol is implemented among the group members. It consists of three stages.

a. To find the Subgroup Controller for each subgroup.

b. CRTDH protocol is used to generate one common key for each subgroup headed by the subgroup controller.

c. Each subgroup gateway member contributes partial keys to generate a one common backbone key (i.e Outer group Key (KG)) headed by the Outer Group Controller using TGDH protocol.

3. Each Group Controller (sub /Outer) distributes the computed public key to all its members. Each member performs rekeying to get the respected group key.

A Regional key KR is used for communication between a subgroup controller and the members in the same region. The Regional key KR is rekeyed whenever there is a membership change event and subgroup joins / leaves and member failure. The Outer Group key KG is rekeyed whenever there is a join / leave subgroup controllers and member failure to preserve secrecy.

The members within a subgroup use Chinese Remainder theorem Group Diffie-Hellman Contributory Key Agreement (CRTDH). Each member within a subgroup contributes his share in arriving at the subgroup key. Whenever membership changes occur, the subgroup controller or previous member initiates the rekeying operation.

The gateway member initiates communication with the neighbouring member belonging to another subgroup and mutually agree on a key using Tree-

Based Group Diffie-Hellman contributory Key Agreement(TGDH) protocol to be used for inter subgroup communication between the two subgroups. Any member belonging to one subgroup can communicate with any other member in another subgroup through this member as the intermediary. In this way adjacent subgroups agree on outer group key. Whenever membership changes occur, the outer group controller or previous group controller initiates the rekeying operation.

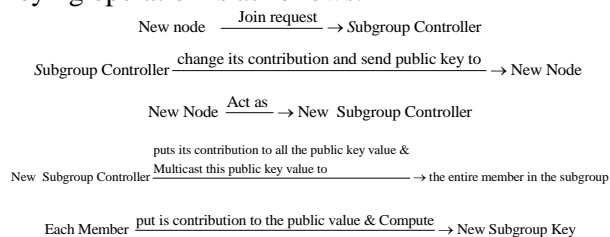
Here, we prefer the subgroup key to be different from the key for backbone. This difference adds more freedom of managing the dynamic group membership. In addition using this approach can potentially save the communication and computational cost.

3.3. Network Dynamics

The network is dynamic in nature. Many members may join or leave the group. In such case, a group key management system should ensure that backward and forward secrecy is preserved.

3.3.1. Member Join

When a new member joins, it initiates communication with the subgroup controller. After initialization, the subgroup controller changes its contribution and sends public key to this new member. The new member receives the public key and acts as a group controller by initiating the rekeying operations for generating a new key for the subgroup. The rekeying operation is as follows.

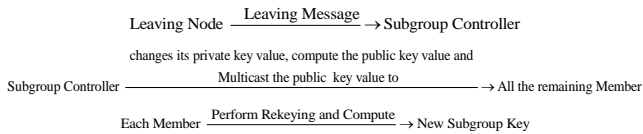


3.3.2. Member Leave:

3.3.2.1. When a Subgroup member leaves

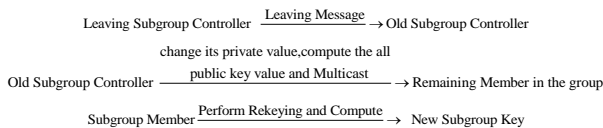
When a member leaves the Subgroup Key of the subgroup to which it belongs must be changed to preserve the forward secrecy. The leaving member informs the subgroup controller. The subgroup controller changes its private key value, computes the public value and broadcasts the public value to all the remaining members. Each member performs rekeying by putting its contribution to public value and computes

the new Subgroup Key. The rekeying operation is as follows.



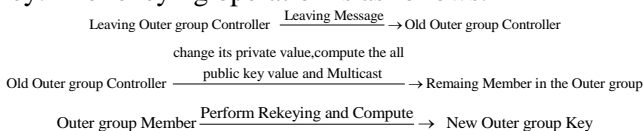
3.3.2.2. When Subgroup Controller Leaves:

When the Subgroup Controller leaves, the Subgroup key used for communication among the subgroup controllers need to be changed. This Subgroup Controller informs the previous Subgroup Controller about its desire to leave the subgroup which initiates the rekeying procedure. The previous subgroup controller now acts as a Subgroup controller. This Subgroup controller change its private contribution value and computes all the public key value and broadcasts to all the remaining member of the group. All subgroup members perform the rekeying operation and compute the new subgroup key. The rekeying operation is as follows.



3.3.2.3. When Outer Group Controller Leaves:

When a Outer group Controller leaves, the Outer group key used for communication among the Outer groups need to be changed. This Outer group Controller informs the previous Outer group Controller about its desire to leave the Outer group which initiates the rekeying procedure. The previous Outer Group controller now becomes the New Outer group controller. This Outer group controller changes its private contribution value and computes the public key value and broadcast to the entire remaining member in the group. All Outer group members perform the rekeying operation and compute the new Outer group key. The rekeying operation is as follows.



3.3.2.4. When Gateway member leaves

When a gateway member leaves the subgroup, it delegates the role of the gateway to the adjacent member having high processing power, memory, and Battery power and acts as a new gateway member. Whenever the gateway member leaves, all the two keys should be changed. These are

- i. Outer group key among the subgroup.

- ii. Subgroup key within the subgroup.

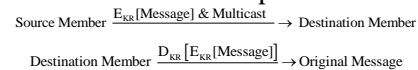
In this case, the subgroup controller and outer group controller perform the rekeying operation. Both the Controller leave the member and a new gateway member is selected in the subgroup, performs rekeying in the subgroup. After that, it joins in the outer group. The procedure is same as joining the member in the outer group.

3.4. Communication Protocol:

The members within the subgroup have communication using subgroup key. The communication among the subgroup members takes place through the gateway member.

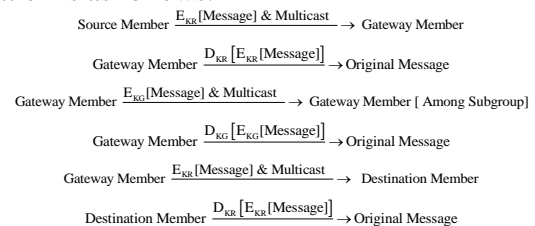
3.4.1. Communication within the Subgroup:

The sender member encrypts the message with the subgroup key (KR) and multicasts it to all member in the subgroup. The subgroup members receive the encrypted message, perform the decryption using the subgroup key (KR) and get the original message. The communication operation is as follows.



3.4.2. Communication among the Subgroup:

The sender member encrypts the message with the subgroup key (KR) and multicasts it to all members in the subgroup. One of the members in the subgroup acts as a gate way member. This gateway member decrypts the message with subgroup key and encrypts with the outer group key (KG) and multicasts to the entire gateway member among the subgroup. The destination gateway member first decrypts the message with outer group key. Then encrypts with subgroup key and multicasts it to all members in the subgroup. Each member in the subgroup receives the encrypted message and performs the decrypts operation using subgroup key and gets the original message. In this way the region-based group key agreement protocol performs the communication. The communication operation is as follows.



4. CRTDH Key Agreement Protocol [2, 3]

4.1. Group Key Establishment

Every group member U_i does as follows.

Step 1.1. Selects a DH private share x_i , computes

$$y_i = g^{x_i} \text{ mod } p.$$

Step 1.2. Broadcast y_i .

Step 1.3. Computes

$$m_{ij} = \begin{cases} y_j^{x_i} \text{ mod } p & \text{if } y_j^{x_i} \text{ mod } p \\ p - y_j^{x_i} \text{ mod } p & \text{otherwise} \end{cases}$$

Step 1.4. For a given $j \neq i$, U_i chooses P_{ij} such that $\text{gcd}(P_{ij}, m_{ij}) = 1$

Step 1.5. For $j \neq i$, U_i uses the following new congruences to replace the original ones:

$$crt_{ij} \equiv k_i \pmod{m_{ij}}$$

$$crt_{ij} \equiv D_i \pmod{P_{ij}}$$

where K_i & D_i are random. U_i also broadcasts the set of pairs

$$crt_i = \{(U_j, crt_{ij}) : j \neq i\},$$

Each group member U_j finds his own matched crt_{ij} ($i \neq j$) from crt_i broadcast by U_i .

Step 1.6. U_i Computes $crt_{ij} \text{ mod } m_{ij}$, $j \neq i$, which must be k_j since $m_{ij} = m_{ji}$, and then computes $GK = k_1 \oplus k_2 \oplus \dots \oplus k_n$, which is the group key.

The illustration of three users CRDH Key establishment is shown in Figure.7 below and The implementation as shown in Figures.16.

USER - I	USER - II	USER - III
G = 5, p = 32713	G = 5, p = 32713	G = 5, p = 32713
Step 1.1: X ₁ = 81234 Y ₁ = 9506	Step 1.1: X ₂ = 96727 Y ₂ = 25139	Step 1.1: X ₃ = 52410 Y ₃ = 21122
Step 1.2: Y ₁ = 9506 Y ₂ = 25139 Y ₃ = 21122	Step 1.2: Y ₁ = 9506 Y ₂ = 25139 Y ₃ = 21122	Step 1.2: Y ₁ = 9506 Y ₂ = 25139 Y ₃ = 21122
Step 1.3: M ₁₂ = 32221 M ₁₃ = 20490	Step 1.3: M ₂₁ = 32221 M ₂₃ = 21291	Step 1.3: M ₃₁ = 20490 M ₃₂ = 21291
Step 1.4: K ₁ = 5876 D ₁ = 5007 P ₁₂ = 16935 P ₁₃ = 16769	Step 1.4: K ₂ = 13390 D ₂ = 11559 P ₂₁ = 3131 P ₂₃ = 5161	Step 1.4: K ₃ = 4387 D ₃ = 3266 P ₃₁ = 2107 P ₃₂ = 15634
Step 1.6: Crt ₁₂ = 296016521377 Crt ₁₃ = 1823983436126 crt ₁ = {(User ₂ , 296016521377), (User ₃ , 1823983436126)}	Step 1.6: Crt ₂₁ = 1329951302826 Crt ₂₃ = 1377419363491 crt ₂ = {(User ₁ , 1329951302826), (User ₃ , 1377419363491)}	Step 1.6: Crt ₃₁ = 155770414237 Crt ₃₂ = 1149926807932 crt ₃ = {(User ₁ , 155770414237), (User ₂ , 1149926807932)}
Step 1.6: K ₁ = 5876 K ₂ = 13390 K ₃ = 4387	Step 1.6: K ₁ = 5876 K ₂ = 13390 K ₃ = 4387	Step 1.6: K ₁ = 5876 K ₂ = 13390 K ₃ = 4387
Group Key GK = k ₁ ⊕ k ₂ ⊕ k ₃ GK = 5876 ⊕ 13390 ⊕ 3387 GK = 13209	Group Key GK = k ₁ ⊕ k ₂ ⊕ k ₃ GK = 5876 ⊕ 13390 ⊕ 3387 GK = 13209	Group Key GK = k ₁ ⊕ k ₂ ⊕ k ₃ GK = 5876 ⊕ 13390 ⊕ 3387 GK = 13209

Fig: 7. CRTDH Key Establishment for Three Users

4.2. The CRTDH join operation

Suppose that U_{n+1} joining the group is shown in Fig.8 below & The implementation is shown in Fig.17.

Step 2.1. U_i ($1 \leq i \leq n$) compute the hash value $h(GK)$. One of them, say U_1 , transmits $h(GK)$ and y_i ($1 \leq i \leq n$) to U_{n+1} .

Step 2.2. U_{n+1} executes Steps 1.1-1.2, executes Step 1.3 only for $m_{n+1,t}$ ($1 \leq t \leq n+1$; $t \neq n+1$), and executes Steps 1.4-1.5 only for $crt_{n+1,t}$ ($1 \leq t \leq n+1$; $t \neq n+1$), i.e., compute and broadcast y_{n+1} and crt_{n+1} .

Step 2.3. U_i ($1 \leq i \leq n+1$) recovers k_{n+1} by using the method in Step 1.6, and computes the new group key as $GK_{new} = h(GK) \oplus k_{n+1}$.

USER - I	USER - IV
Received the join request from new User -4	Step 1: X ₄ = 47201 Y ₄ = 13475
Step 2.1: All current member User1, User2 and User3 should compute the hash value of the current group key. h(GK) = h(13209) h(13309) = 440665184142888518086951774487012380961039488 User -1 send the following information to user-4 Y ₁ = 9506, Y ₂ = 25139, Y ₃ = 21122 and h(13309) = 440665184142888518086951774487012380961039488	Step 2: Y ₁ = 9506, Y ₂ = 25139, Y ₃ = 21122 and h(13309) = 440665184142888518086951774487012380961039488
Step 2.2: M ₁₄ = 26525 crt ₁₄ = {(User ₁ , 1056184071993), (User ₂ , 1125765570445), (User ₃ , 22324435823)}	Step 3: M ₄₁ = 2525 M ₄₂ = 29578 M ₄₃ = 21795
K ₄ = 4043	Step 4: K ₄ = 4043 D ₄ = 2763 P ₄₁ = 10386 P ₄₂ = 10663 P ₄₃ = 2732
Step 2.3: GK = h(GK) ⊕ k ₄ GK = 440665184142888518086951774487012380961039488 ⊕ 4043 GK = 4406651841428885180869517744870123809610395651	Step 5: crt ₄ = {(User ₁ , 1056184071993), (User ₂ , 1125765570445), (User ₃ , 22324435823)}
	Step 6: GK = h(GK) ⊕ k ₄ GK = 440665184142888518086951774487012380961039488 ⊕ 4043 GK = 4406651841428885180869517744870123809610395651

Figure: 8. User 4 Join the Group

4.3. The CRTDH leave operation

Suppose that $n > s > 1$ & U_s is going to leave given in Fig.9 below & The implementation is shown in Fig.18.

Step 3.1. One of the U_i , say U_1 , repeats Steps 1.4 -1.5 with a new k_1 , broadcasts the new $crt_1 = \{crt_{1,t} : 2 \leq t \leq n\}$, & computes the new group key $GK_{new} = GK \oplus k_1$.

Step 3.2. U_i ($2 \leq i \leq n$) recovers k_1 from crt_1 and then compute the new group key $GK_{new} = GK \oplus k_1$.

USER - IV
Assume, User 4 received the Leave request from User 2
Step 3.1: User 4 change its random value K ₄ and D ₄ K ₄ = 15980 D ₄ = 1665 P ₄₁ = 17067 P ₄₃ = 41
Step 3.2: crt ₁ = {(User ₁ , 2183719579605), (User ₃ , 10535718980)} h(GK) = 377826141864761023844535532961914335967304333992 GK = h(GK) ⊕ k ₄ GK = 377826141864761023844535532961914335967304333992 ⊕ 15980 GK = 3778261418647610238445355329619143359673043320196

Figure.9. After User 2 leave the group

5. TGDH Protocol [4, 5, 6 & 7]

One of the main features of our work is the use of key trees in fully distributed contributory key agreement. Figure.10 shows an example of a key tree. The root is located at level 0 and the lowest leaves are at level h . Since we use binary trees, every node is either a leaf or a parent of two nodes. The nodes are denoted $\langle l, v \rangle$, where $0 \leq v \leq 2^l - 1$ since each level l

hosts at most 2^l nodes. Each node $\langle l, v \rangle$ is associated with the key $K_{\langle l, v \rangle}$ and the blinded key (bkey) $BK_{\langle l, v \rangle} = f(K_{\langle l, v \rangle})$, where the function $f(\cdot)$ is modular exponentiation in prime order groups, that is, $f(k) = \alpha^k \text{ mod } p$ (equivalent to the Diffie–Hellman protocol).

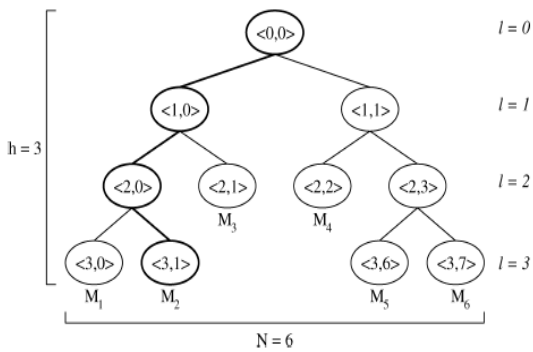


Figure 10. A Key Tree.

5.1 Join Protocol

Assume the group has n members: $\{M_1, \dots, M_n\}$. The new member M_{n+1} initiates the protocol by broadcasting a join request message that contains its own bkey $BK_{\langle 0,0 \rangle}$. Each current group controller receives this message and determines the insertion point in the tree. The insertion point is the shallowest rightmost node, where the join does not increase the height of the key tree. Otherwise, if the key tree is fully balanced, the new member joins to the root node. The group controller is the rightmost leaf in the sub tree rooted at the insertion node. The group controller proceeds to update its share and passed all bkeys tree structure to new joining member. The new joining member acts as the group controller and computes the new group key. Next, the group controller broadcasts the new tree that contains all bkeys. All other members update their trees accordingly and compute the new group key.

5.1.1 . Illustrating with examples

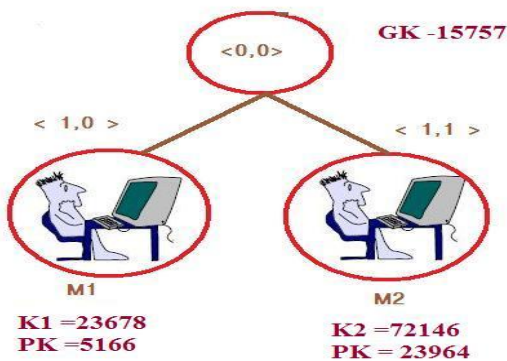


Figure: 11. User M1&M2 Join using TGDH

User M_1 and User M_2 are going to exchange their keys: Take $g = 5$ and $p = 32713$. User M_1 's private key is **23678**, so M_1 's public key is **5166**. User M_2 's private key is **72146**, so M_2 's public key is **239640**. The Group key is computed (Figure.11) as User M_1 sends its public key **5166** to user M_2 , the User M_2 computes their group key as **15757**. User M_2 sends its public key **23964** to user M_1 , then the user M_1 computes their group key as **15757**. Here, Group controller is User M_2 .

When 3rd node Joins

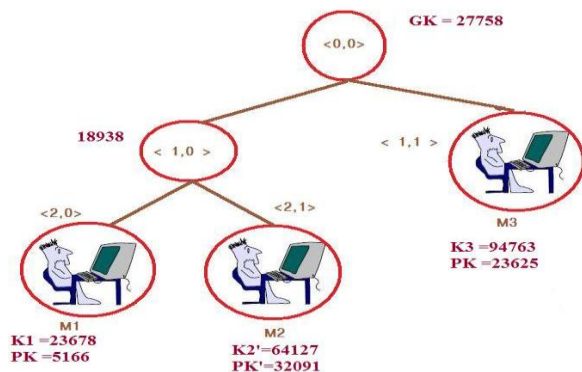


Figure 12. User M_3 Join the Group

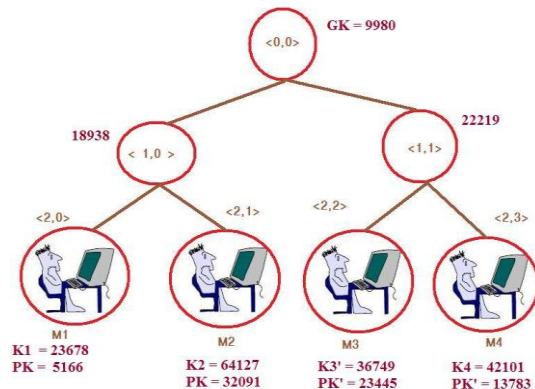


Figure. 13. User M_4 Join the group

When User joins the group, the old group controller M_2 changes its private key value from **72146** to **64127** and passes the public key value and tree to User M_3 . Now, M_3 becomes group controller. Then, M_3 generates the public key **23625** from its private key as **94763** and computes the group key as **27758** shown in Figure.12. M_3 sends Tree and public key to all users. Now, user M_1 and M_2 compute their group key. The same procedure followed by joining the User 4 as shown in Fig. 13. The implementations are as shown in Figures.19, 20&21.

5.2. Leave Protocol

There are two types of leave, 1. Ordinary member leave and 2. Group controller leave

5.2.1. Ordinary member leave

When user M_2 leaves (Figure.14) the group, then the Group controller changes its private key **42101** to **27584** and group key is recalculated as **7914**. After that, it broadcast its Tree and public key value to all users in the group. Then, the new group key will be generated by the remaining users. The implementation is as shown in Fig.22.

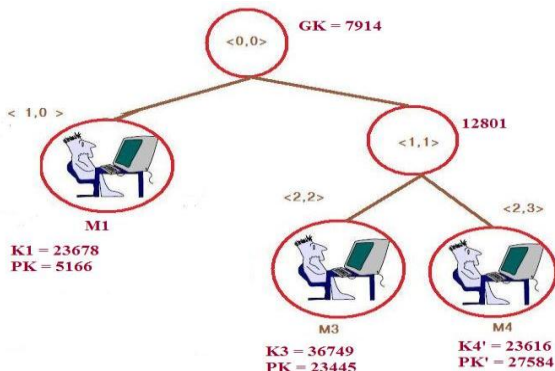


Figure.14. User M_2 Leave from the Group

5.2.2. When a Group controller leaves

When a Group controller leaves (Figure.15) from the group, then its sibling changes its private key value 36749 to **14214** and recalculates the group key as **6576**. After that, the same steps are followed by ordinary member leave method. The implementation is as shown in Figure.23.

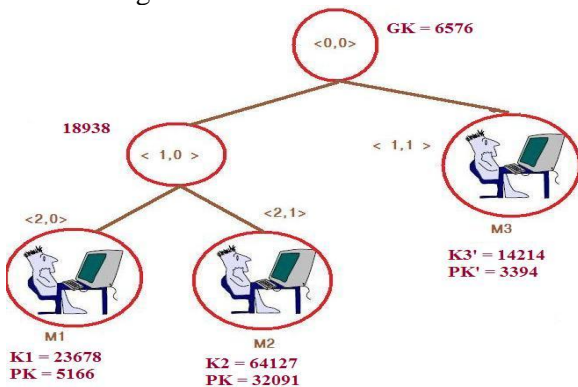


Figure.15. Group Controller Leave from the Group

6. Experimental Results and Discussion

The experiments were conducted on sixteen Laptops running on a 2.4 GHz Pentium CPU with 2GB of memory and 802.11 b/g 108 Mbps Super G PCI wireless cards with Atheros chipset. To test this project in a more realistic environment, the implementation is

done by using Net beans IDE 6.1, in an ad-hoc network where users can securely share their data. This project integrates with a peer-to-peer (P2P) communication module that is able to communicate and share their messages with other users in the network which is described below.

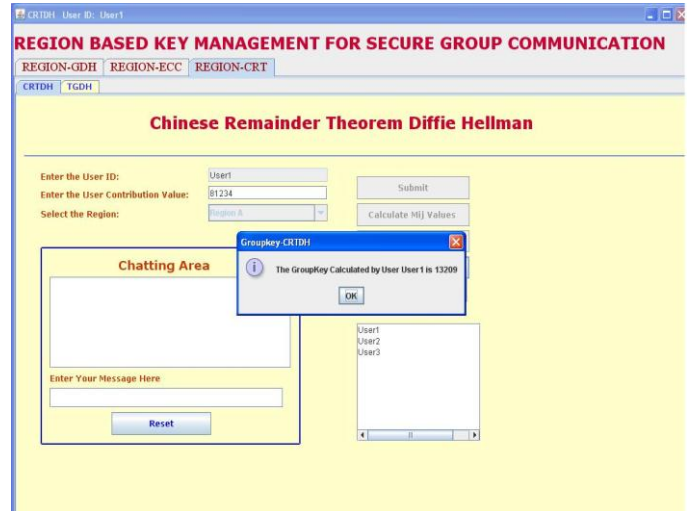


Figure.16. Group Key of User 1, 2 & 3

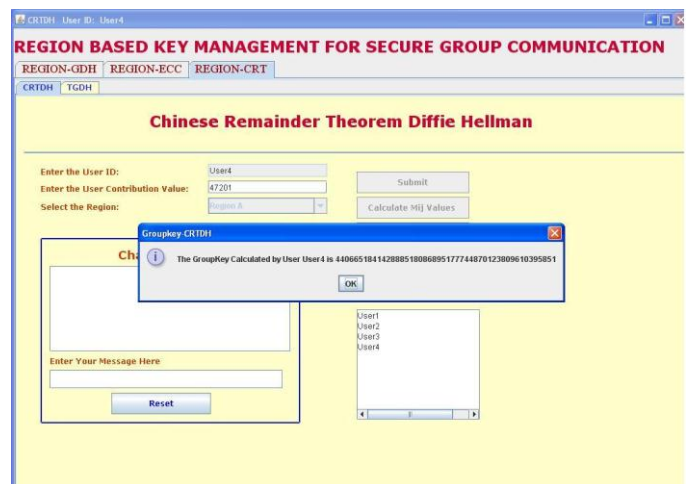


Figure.17. Group Key after User4 Join.

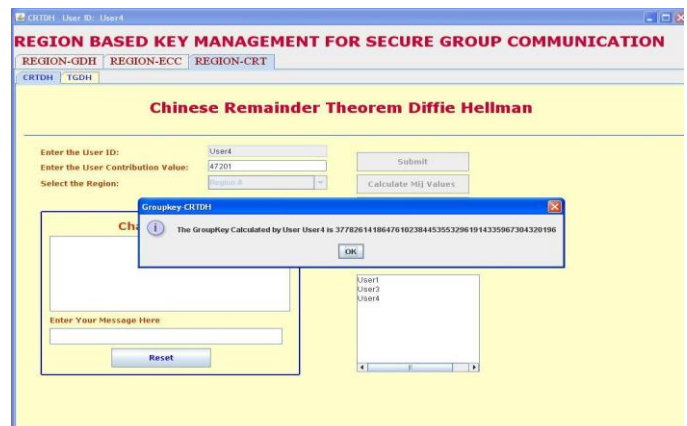


Figure.18. Group Key after User2 Leave

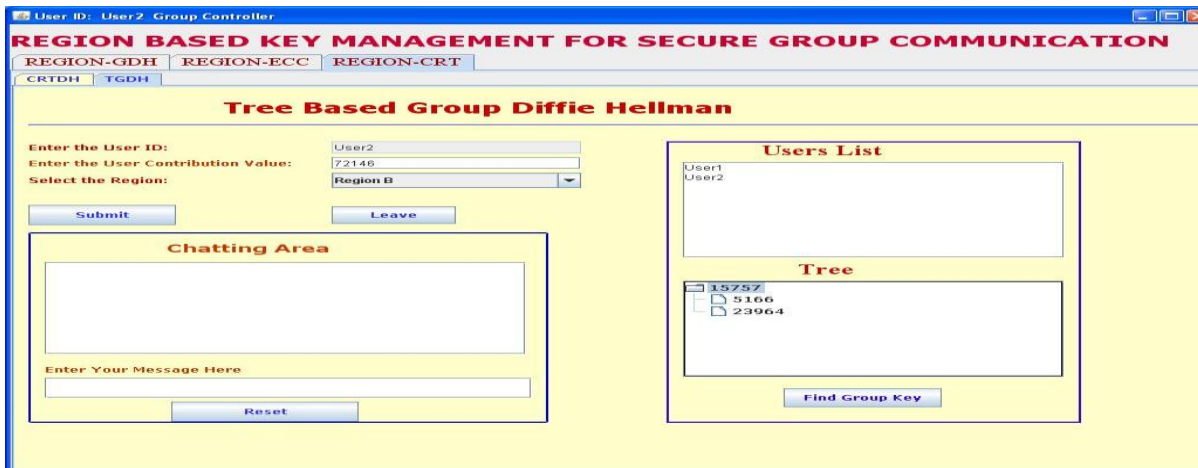


Figure19. Group Key of User M_1 & M_2

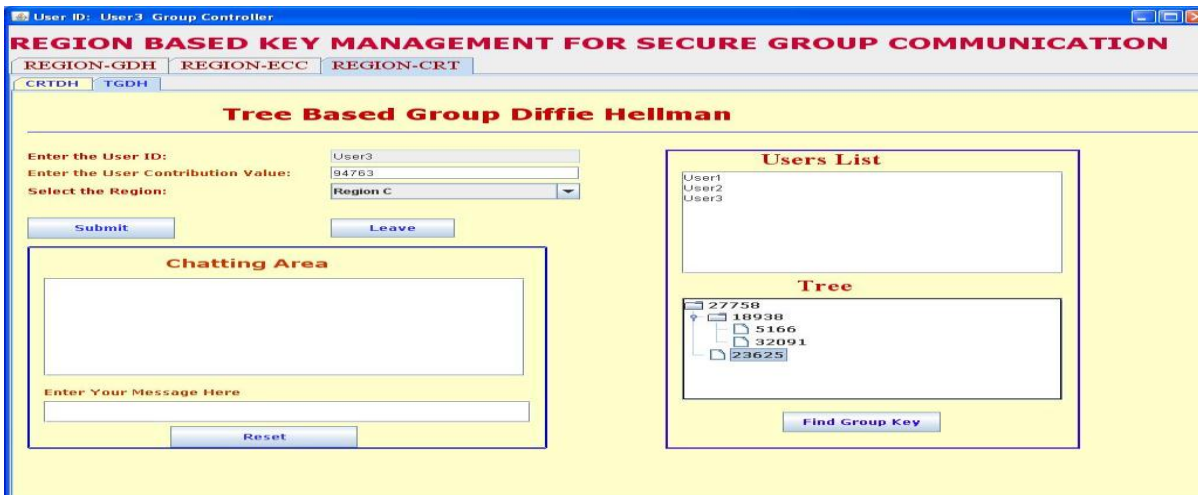


Figure 20. Group Key of User M_1, M_2 & M_3 M_1 & M_2



Figure 21. Group Key of User M_1, M_2, M_3 & M_4



Figure. 22. Group Key after M_2 Leave

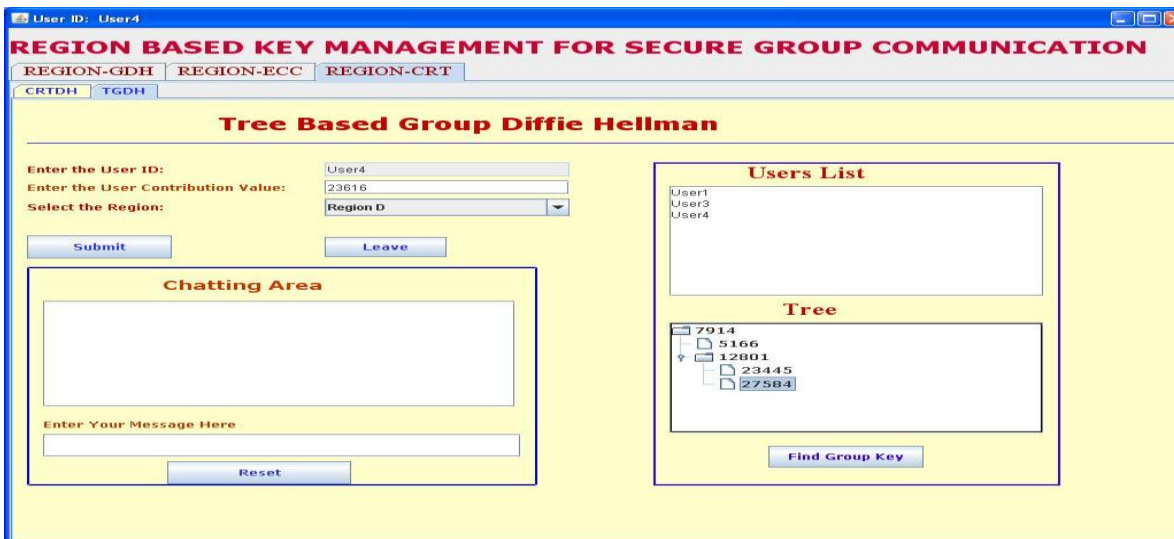


Figure.23. Group Key after Group Controller Leave

7. Complexity Analysis

7.1. Memory Costs:

Our approach consumes very less memory compared to TGDH and CRTDH when members go on increasing.

7.2. Communication Costs:

Our approach consumes less bandwidth when compare to CRTDH and TGDH. TGDH depends on trees height, balance of key tree, location of joining tree, and leaving nodes. But this approach depends on the number of members in the subgroup, number of Group Controller, and height of tree.

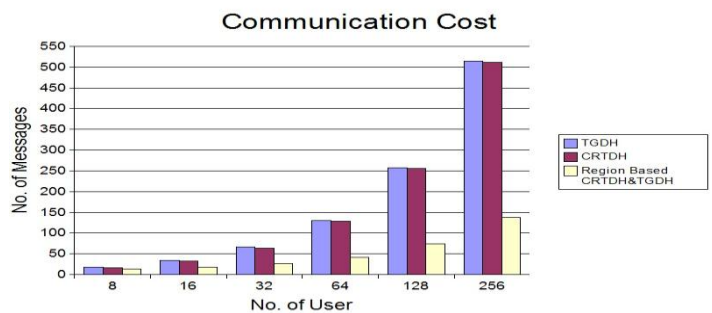


Figure.24. Communication Cost

Considering (Figure.24) there are 256 members in a group our approach consumes only 29% of Bandwidth when compare to CRTDH and TGDH.

7.3. Computation Costs:

The Computational costs depend on the Number of exponentiations. CRTDH has high computation costs as it depends on the number of members and group size respectively. The cost increases as the members and group size increases. But our approach spends a little on this computation.

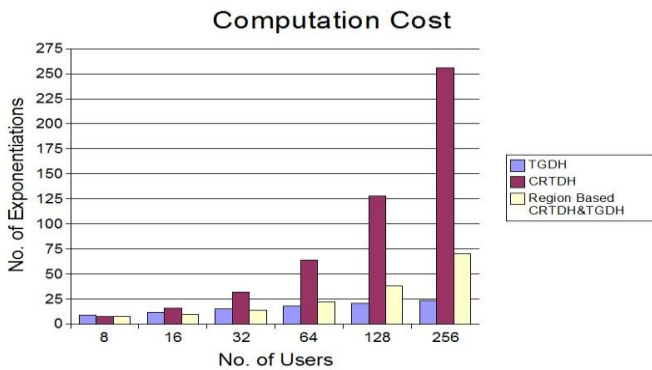


Figure.25. Computation Cost

Consider (Figure.25) there are 256 members in a group. Our approach consumes nearly 27% of less exponentiation when compared to CRTDH. But in this case TGDH is less compared to our approach. But this occurs because of additive nature of CRTDH and TGDH. Performance wise our approach leads the other two methods to overcome member serialization, even for the very large groups.

8. Conclusion

In this paper, a region-based key agreement scheme has been proposed and implemented, which can enhance the secure group communication performance by using multiple group keys. In contrast to other existing schemes using only single key, the new proposed scheme exploits asymmetric key, i.e an Outer group Key and multiple Subgroup keys, which are generated from the proposed Region-Based key agreement algorithm. By using a set comprising an outer group key and subgroup keys a region-based scheme can be efficiently distributed for multiple secure groups. Therefore, the number of rekeying messages, computation and memory can be dramatically reduced. Compared with other schemes,

the new proposed Region-Based scheme can significantly reduce the storage and communication overheads in the rekeying process, with acceptable computational overhead. It is expected that the proposed scheme can be the practical solution for secure group applications, especially for Battlefield Scenario.

References

- [1]. Kumar. K , Sumathy. V and J. Nafeesa Begum, "Efficient Region-Based Group Key Agreement protocol for Ad Hoc networks using Elliptic Curve Cryptography", IEEE International Advance computing Conference, 6-7 March 2009, pp.1052-1060.
- [2]. Spyros Magliveras, Wandu Wei and Xukai Zou, "Notes on the CRTDH Group Key Agreement Protocol", The 28th International Conference on Distributed Computing Systems Workshops, 2008, pp.406-411.
- [3]. R.Balachandran, B.Ramamurthy, X.Zou and N.Vinod Chandran, "CRTDH: An efficient key agreement scheme for secure group communications in wireless ad hoc networks", Proceedings of IEEE international Conference on Communications (ICC), 20(8), pg.1123-1127, 2005.
- [4] Amir.Y, Kim.Y, Nita-Rotaru.C, Schultz.J, J.Stanton, and Tsudik.G , "Exploring Robustness in Group Key Agreement," Proc. 21st IEEE Int'l Conf. Distributed Computing Systems, pp. 399-408, Apr.2001.
- [5] Patrick P.C.Lee, John C.S.Lui and David K.Y. Yau, "Distributed Collaborative Key Agreement and Authentication Protocols for Dynamic Peer Groups," IEEE/ACM Transactions on Networking, Vol. 14, No.2, April.2006.
- [6] Steiner.M, Tsudik.G, and Waidner.M, "Key Agreement in Dynamic Peer Groups", IEEE Trans. Parallel and Distributed Systems, vol. 11, no.8, Aug.2000.
- [7] Yair Amir, Yongdae Kim, Cristina Nita-Rotaru, John L.Schlitz, Jonathan Stanton and Gene Tsudik, "Secure Group Communication Using Robust Contributory Key Agreement", IEEE, vol.15, no.5, May 2004.
- [8] William Stallings, "Cryptography and network security principles and practices", Third Edition, Pearson Education

A Migrating Parallel Exponential Crawling Approach to Search Engine

Jitendra Kumar Seth

Assistant Professor, Department of Information
Technology, Ajay Kumar Garg Engg. College, Ghaziabad,
India, mrjkseth@yahoo.co.in

Ashutosh Dixit

Senior Lecturer in Computer Science Department,
YMCA, Faridabad, Hariyana, India
dixit_ashutosh@rediffmail.com

Abstract. Search engines have become important tools for Web navigation. In order to provide powerful search facilities, search engines maintain comprehensive indices of documents available on the Web. The creation and maintenance of Web indices is done by Web crawlers, which recursively traverse and download Web pages on behalf of search engines. Analysis of the collected information is performed after the data has been downloaded. In this research, we propose an alternative, more efficient approach to building parallel Web crawlers based on mobile crawlers. Our proposed crawlers are transferred to the remote machines where they download the web pages and make some other processing in order to filter out any unwanted data locally before transferring it back to the search engine (central machine). This reduces network load and speeds up the indexing phase inside the search engine. In our approach design and implementation of web crawler that can grow parallelism to the desired depth and can download the web pages that can grow exponentially in size is being proposed. The parallel crawler first filters and then compresses the downloaded web pages locally before transmitting it to the central machine. Thus the crawlers save the bandwidth of network and take the full advantage of parallel crawling for downloading and speedup the process.

Keywords: Quality of index, search engine, web crawlers, parallel crawlers, mobile crawlers, communication bandwidth, Crawl machine, CDB, Crawler Application, filter, and compress.

1. Introduction

The World Wide Web ("WWW" or simply the "Web") is a global information medium which users can read and write via computers connected to the Internet. It is not easy to find your web pages among 1 billion web pages currently published online. The size of the Web has doubled in less than two years, and this growth rate is projected to continue for the next two years. In the context of Internet for useful information, a search engine is a program, or series of programs that, scan and index the web pages on the internet. A crawler is a program that retrieves and stores pages from the Web, commonly for a Web search engine. A crawler often has to download hundreds of millions of pages in a short period of time and has to constantly monitor and refresh the downloaded pages. Roughly, a crawler starts off

by placing an initial set of URLs, in a queue, where all URLs to be retrieved are kept and prioritized. From this queue, the crawler gets a URL (in some order), downloads the page, extracts any URLs in the downloaded page, and puts the new URLs in the queue. This process is repeated. Collected pages are later used for other applications, such as a Web search engine.

Given this explosive growth, we see the following specific problems with the way current search engines index the Web:

Scaling: The concept of "download-first-and-index-later" will likely not scale given the limitations in the infrastructure and projected growth rate of the Web. Using the estimates for growth of Web indices provided in [1], a Web crawler running in the year 2000 would have to retrieve Web data at a rate of 45Mbit per second in order to download the estimated 480GB of pages per day that are necessary to maintain the index. Looking at the fundamental limitations of storage technology and communication networks, it is highly unlikely that Web indices of this size can be maintained efficiently.

Efficiency: Current search engines add unnecessary traffic to the already overloaded Internet. While current approaches are the only alternative for general-purpose search engines trying to build a comprehensive Web index, there are many scenarios where it is more efficient to download and index only selected pages.

Quality of Index: The results of Web searches are overwhelming and require the user to act as part of the query processor. Current commercial search engines maintain Web indices of up to 110 million pages [1] and easily find several thousands of matches for an average query. Thus increasing the size of the Web index does not automatically improve the quality of the search results if it simply causes the search engine to return twice as many matches to a query as before.

Since we cannot limit the number of pages on the Web, we have to find ways to improve the search results in such a way that can accommodate the rapid growth of the Web.

Therefore, we expect a new generation of specialized search engines to emerge in the near future.

2. Literature Survey

According to Junghoo Cho, Hector Garcia-Molina[2], many search engines often run multiple processes in parallel to perform the task of parallel crawling, so that download rate is maximized. In particular, following issues make the study of a parallel crawler challenging and interesting:

Overlap: When multiple processes run in parallel to download pages, it is possible that different processes download the same page multiple times.

Quality: Often, a crawler wants to download “important” pages first, in order to maximize the “quality” of the downloaded collection. However, in a parallel crawler, each process may not be aware of the whole image of the Web that they have collectively downloaded so far.

Communication bandwidth: In order to prevent overlap, or to improve the quality of the downloaded pages, crawling processes need to periodically communicate to coordinate with each other. However, this communication may grow significantly as the number of crawling processes increases.

According to Jan Fiedler and Joachim Hammer [3], in the mobile based web crawling approach, the mobile crawler move to data source before the actual crawling process is started. The use of mobile crawlers for information retrieval requires an architecture which allows us to execute code (i.e. crawlers) on remote systems.

A critical look at the available literature indicates the following issues to be addressed towards design of an efficient crawler.

- The traditional parallel web crawlers download the web pages on a single machine which causes bottleneck at the network level.
- The traditional migrating crawlers do not perform any compression and filtering before transmitting the web pages to the central machine. Moreover the migrating web crawlers generally migrates themselves up to a single level of migration depth hence they are unable to take benefit of desired level of migration depth hence reduces the degree of parallelism.

3. A migrating parallel exponential web crawling approach

In this work the concept of download first transmits later is being proposed by using which data can be locally downloaded, filtered and compressed before transmitting it to the search engine server. Design and implementation of parallel migrating web crawler that can grow parallelism to the desired depth and can download the web pages that can grow exponentially in size is being proposed. The proposed parallel migrating crawler also first filters and then compresses the web pages locally before any transmission of file to the central machine (search engine). Thus, the crawler saves the bandwidth of channel and takes the full advantage of parallel crawling and speedup the process.

Architecture of migrating parallel exponential web crawler: The architecture of the proposed crawler is shown below in figure 1.1. Parallel migrating crawler with exponential growth consists of the two major modules

- Web crawler application
- Crawl machine module

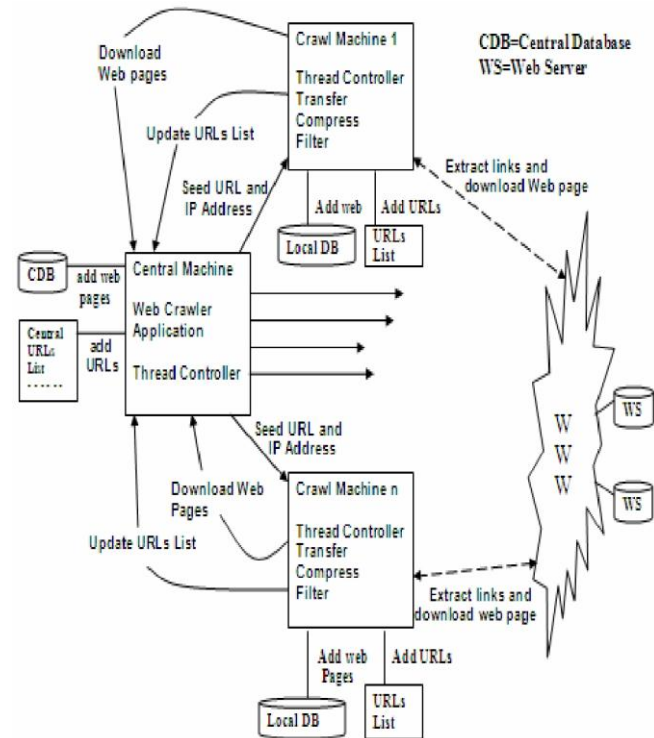


Fig 1.1 Architecture of Parallel migrating crawler with exponential growth

In continuation of the figure 1.1 the next level of crawl machine modules running on n number of machines are shown in figure 1.2.

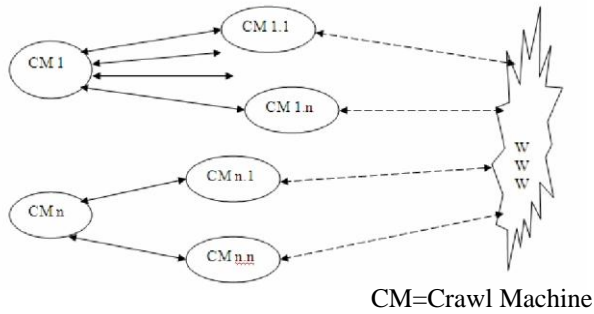


Fig 1.2 Architecture of Parallel migrating crawler with exponential growth

Web crawler application (search engine or central machine):

This is the main module that represents the central machine or more generally search engine, prompts the user to enter the seed URLs to be crawl and the IP addresses of the machine to which the crawling process will migrate. It is a multithreaded module that creates the required numbers of thread of crawling processes and controls these processes on remote machines. This module dynamically updates its central database of downloaded web pages getting from different crawling threads running on remote machines and central URLs list based on current crawling URLs by different crawling processes running on remote machines.

Each crawling process downloads web pages and stores them in its local database that is referred to a directory named crawlerdatabase. The central machine stores downloaded web pages received from different crawling processes (crawl machine module) on remote machines into its central database that is a directory. Each remote machine crawling process also acts as a central machine generates the threads of crawling processes and migrate them on different remote machines and the next level remote machines running the crawling process also acts as a central machine and so on. The level of migration increases until we achieve the satisfactory level of parallelism in crawling which improves the performance of overall crawling.

This module consists of:

- o Counter_wca

- o URLlistupdate_wca
- o pageDownloader_wca
- o ThreadController_wca

Counter_wca

The counter module is used to control the depth of crawling. This module initializes a counter variable to the desired number of depth of crawling. Then it sends the counter value to each next level crawl machines. Each crawl machine decrements the counter value by one and forwards the updated counter value to the next level crawl machines. The next level crawl machine decrements the counter value by one and forwards the updated counter value to the next level crawl machine and so on this process of counter forwarding continues until the counter value reaches to zero. As the next level crawl machine find the counter value zero it stops migrating crawling process to the next level.

Algorithm Counter_wca

```

Begin
    Initialize a counter variable to
    the desired number of depth of
    crawling; Send the counter value to
    the next level of crawl machines;
End
    
```

URLlistupdate_wca

This module is responsible for updating the current crawling status of each crawl machine visualizing on the central machine. It also updates the central URL list which is also visualized in a synchronized manner at the central machine. As soon as a new URL is found by any crawl machine it gets the URL and adds the URL to the central URL list.

Algorithm URLlistupdate_wca

```

Begin
    While (not receiving "done" from the
    crawl machine)
    {
    
```

```
Receive URL from any
crawl machine module;

Display crawling status;
Update the corresponding
crawl machine status at
the central machine;

If URL can not resolve then
    Display status and do
    not add URL in central
    URL list;
Else
    Add URL to the central URL
    List;
}
```

End

pageDownloader_wca

The crawl machines downloads the web pages from the web and then filter the web pages based on some user choice and then compress the filtered web pages in a zip file and finally transfer the zip file to the central machine. This module is responsible for download the zip files from the crawl machines and store the downloaded zip files in the central local 'crawlerdatabase' directory with synchronized access.

Algorithm pageDownloader_wca

Begin

```
While receiving zip files
{
    Create zip file with the same
    name in crawlerdatabase
    directory as transferring Crawl
    machine's (1...n) zip file name;
    Receive contents of zip file;
    Write contents to the zip file
    ; Display zip file received ;
    Next zip file;
}
```

End

This module is also implemented as a part of each crawl machine module.

ThreadController_wca

The thread controller module generates a number of threads set by the programmer. Each thread is responsible to migrate the crawling process to the destination machine and also control the data transfer between the WebCrawler applications and crawl machines. The thread controller module also supplies the seed urls to the crawl machines. The URL list update module and page downloader modules are the part of the thread controller. The thread controller module also synchronizes the access to the central crawlerdatabase directory and the

Algorithm ThreadController_wca

Begin

```
Create desired number of threads
of crawling Process;

Assign a seed URL and IP address of
the crawl machine to each thread;
Each thread makes connection to the
Appropriate crawl machine;

Migrate the crawling process to
the destination crawl machine;
Synchronize and control the
transmission between the web
crawler application and Crawl
machines using URLlistupdate_wca
and pagedownloader_wca;
```

End

Crawl Machine Module

Each crawl machine modules are the process that takes the different seed URLs from the central machine and each crawl machine crawl the web independently. Each crawl machine are also capable to act as a central machine that can connect and supply the seed URLs to other next crawling level independent crawl machines which in turn can connect and supply seed URLs to other next crawling level independent crawl machines and so on, which causes it can grow the crawling size exponentially. The crawl machines extract the URLs links from the web pages and update the local URL list and parallel update the URL list of

the central machine. Each crawl machine also downloads the web pages from the web and updates its local page database.

The crawl machine module consists of the following components-

- o Counter_cm
- o ThreadController_cm(Same as ThreadController_wca)
- o pageDownloader_cm
- o Filter_cm
- o Compress_cm
- o Transfer_cm

Counter_cm : This module is similar to Counter_wca.

Algorithm Counter_wca

Begin

```
Decrement the received counter value by 1;
If counter value=0 then
    Don't migrate the crawling process to next level;
Else
    Send the counter value to the next level of crawl machine;
```

End

pageDownloader_cm

The page downloader module crawls the web starting with the seed URL supplied by the thread controller module of the previous level crawl machine or from the central machine. It uses four data structures vectortosearch, vectorsearched, vectormatches and a URL List. It stores each new extracted URL from web pages at the last of vectortosearch, vectormatches and the URL list. Each time it takes a URL in FIFO order from the vectortosearch once the URL is resolved and crawled on the web it deletes the URL from the vectortosearch and add this URL at the last of vectorsearched. When this module finds URL from the web pages it checks the vectormatches for the URL, if this URL is already in vectormatches then the URL is ignored otherwise it is added to the last of vectotosearch, vectormaches and URL List.

Algorithm pageDownloader_cm

Begin

```
While (vectortosearch is not empty or crawl size reaches to the max. limit)
{
    Take the URL from vectortosearch in FIFO Order;
    If URL is not robot safe then Break;
    If URL can not resolve then Break;
    Extract the file on the URL;
    Store the file in the crawlerdatabase; While there is a hyperlink in the file
    {
        Extract the URL from the file;
        If URL can not resolved then
            Write message to the invoking process thread;
            Continue;
        If URL is not robot safe then Write message to the invoking application thread;
        Continue;
        If URL is not in vectormatches then Add URL to vectortosearch, vectormatches, and URL List;
        Send URL to the invoking process thread;
    }
    Increase crawl size by 1;
}
Send message "done" to the invoking thread; End
```

Filter_cm

This module filters the web pages in the crawlerdatabase directory based on some user choice. In this work filtering is done based on file extensions such as .html, .txt etc. However files can be filtered as per requirements. This module picks up the filtered files from the crawlerdatabase and stores in a different directory named "filtered" on the same machine local disk drive.

Algorithm

```
filter_cm Begin
    Create a directory "filtered" on
    the local disk drive;
    Read crawlerdatabase directory;
    While there is .html file in
    crawlerdatabase
        Add .html files to the
        directory "filtered";
End
```

Compress_cm

This module is responsible for compressing all the files in the filtered directory into a single .zip file. This module first creates a directory named "zip_n" on local disk drive and then creates a .zip file into the zip_n directory then reads all the files in the filtered directory and then adds all the files into a single .zip file.

Algorithm compress_cm

```
Begin
    Create a directory named zip_n
    on local disk drive;
    Create a .zip file in zip_n directory
    ; While there are files into
    "filtered" directory
        Add files to .zip file;
End
```

Transfer_cm

This module transfer the .zip files in the zip_n directory getting after compression to the previous level crawl machine's zip_n directory or finally to the central machine's crawlerdatabase directory.

Algorithm transfer

```
Begin
    While there are files in
    zip_n directory
    {
        Send zip file to the previous
        level crawl machine's zip_n
        directory or at the last
        level to the central
        machine's crawlerdatabase
        directory; Send contents of
        zip files; Next zip file;
    }
End
```

4. Performance

Focus was on comparing the performance of parallel migrating web crawler with exponential growth with a standalone conventional crawler that does not use migration. This standalone crawler will download the pages locally and does not make any crawling process migration or transmission of data to any other machine as opposed to parallel migrating web crawler with exponential growth that will migrate the crawling process to other machines at some desired crawling depth and finally all crawling machines transmits the web pages to the central machine.

Time Measurement

After the execution of the conventional crawler and the parallel migrating crawler with exponential growth on five URLs sets, the following observations were made:

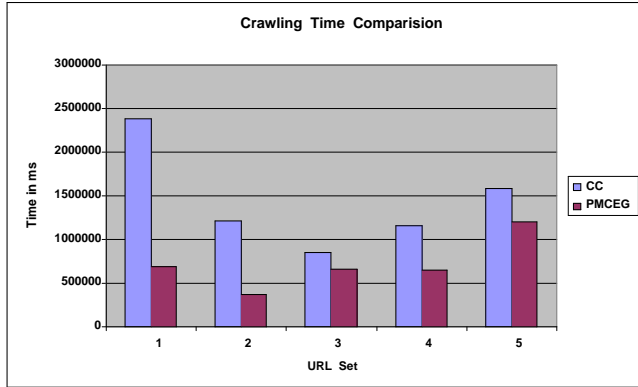


Fig 1.3 Crawling time comparison of Conventional crawler and parallel migrating web crawler with exponential growth

CC= Conventional crawler

PMCEG= Average crawling time of parallel migrating crawler with exponential growth

Average crawling time of Conventional crawler= 1439693.60 milliseconds.

Average crawling time of parallel migrating crawler with exponential growth= 715637.2 milliseconds.

% Benefit in time= $100 - ((715637.2/1439693.6) * 100) = 51$

Quality Measurement

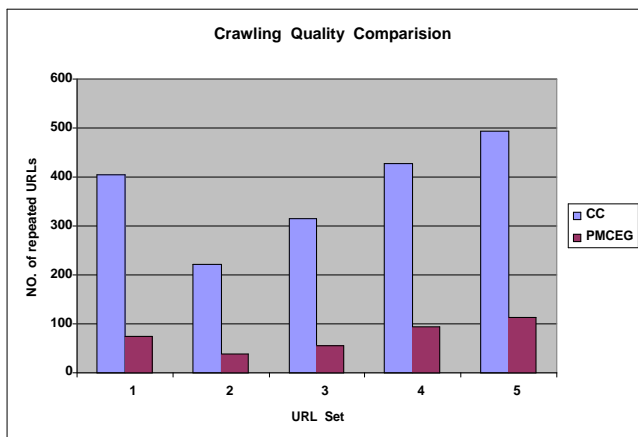


Fig 1.4 Crawling quality comparison of Conventional crawler and parallel migrating web crawler with exponential growth

Average number of repeated URLs in Conventional crawler= 372.2

Average number of repeated URLs in parallel migrating crawler with exponential growth= 74.6

%Benefit in quality= $100 - ((74.6/372.2) * 100) = 80$

Network Resource Utilization

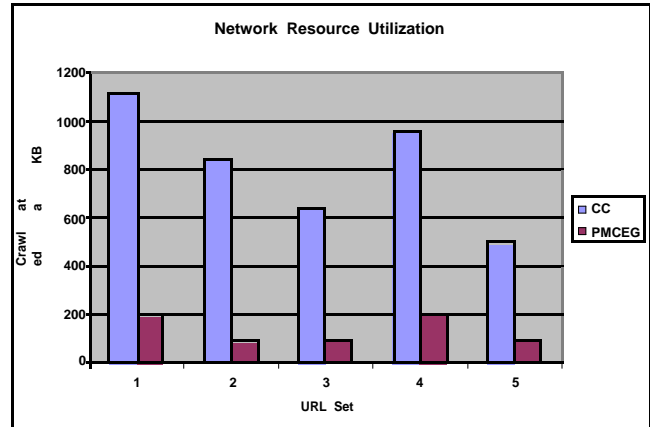


Fig 1.5 Network resource utilization comparison of Conventional crawler and parallel migrating web crawler with exponential growth

Average Crawled data in Conventional crawler= 810.68 KB

Average Crawled data in parallel migrating crawler with exponential growth= 136.62 KB

%Benefit in network bandwidth=

$100 - ((136.62/810.68) * 100) = 83$

5. Conclusion

The traditional parallel web crawlers download the web pages on a single machine, which causes bottleneck at the network level. The traditional migrating crawlers do not perform any compression and filtering before transmitting the web pages to the central machine. Moreover the Migrating web crawlers generally migrates themselves up to a single level of migration depth hence they are unable to take benefit of desired level of migration depth hence reduces the degree of parallelism. In this work the concept of download first transmits later is being proposed by using which data can be locally downloaded, filtered and compressed before transmitting it to the search engine server. In this work design and implementation of web crawler that can grow parallelism to the desired depth and can download the web pages that can grow exponentially in size is being proposed. The parallel crawler first filters and then compresses the downloaded web pages locally before

Transmitting it to the central machine. Thus the crawlers save the bandwidth of network and take the full advantage of parallel crawling for downloading and speedup the process.

6. Future work

In this work the following future aspects are arising. While downloading the web pages, we are using socket programming for file data transmission and dynamic URL list update it can be improved by using ftp protocol to reduce the time of file transmission. The socket programming produces the more number of lines of codes, the code optimization can be done by using some other technologies like RMI, JSP or servlet.

We have not provided any security measures in system connections, crawling process and in transmission of data among machines. It can be implemented some security mechanism.

The crawling process does not migrate automatically to the other machines we have to run the crawling machine module manually on other machines that will receive the crawling process by the central machine. This migration can be automatic using some other techniques like agelet etc.

References

1. Sullivan, D., Search Engine Watch, Mecklermedia, 1998, <http://www.searchenginewatch.com>.
2. Junghoo Cho, Hector Garcia-Molina, Parallel Crawlers WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA. ACM 1-58113-449-5/02/0005
3. Joachim Hammer and Jan Fiedler, Using Mobile Crawlers to Search the Web Efficiently. UF Technical Report, Number TR98-007, Gainesville, FL, June 1998
4. Douglas E. Comer, "The Internet Book", Prentice Hall of India, New Delhi, 2001.
5. Francis Crimmins, "Web Crawler Review".
6. A.K. Sharma, J.P. Gupta, D. P. Aggarwal, PARCAHYDE: An Architecture of a Parallel Crawler based on Augmented Hypertext Documents.
7. A.K.Sharma, Charu Bishnoi, Dimple Juneja, "A Multi-Agent Framework for agent based focused crawlers", Proc. Of International Conference on Emerging Technologies in IT Industry, pp- 48, ICET-04, Punjab, India, November 2004.
8. Junghoo Cho and Hector Garcia-Molina. Estimating frequency of change, 2000. Submitted to VLDB 2000, Research track.
9. [9 Sriram Raghavan Hector GarciaMolina, Crawling the Hidden Web, Computer Science Department Stanford University Stanford, CA 94305, USA
10. The Deep Web: Surfacing Hidden Value. <http://www.completeplanet.com/Tutorials/DeepWeb>.
11. S. Lawrence and C. L. Giles. Searching the World Wide Web. Science, 280(5360):98, 1998.
12. http://en.wikipedia.org/wiki/Web_crawler
13. S. Lawrence and C. L. Giles. Accessibility of information on the web. Nature, 400:107{109, 1999.
14. S. Raghavan and H. Garcia-Molina. Crawling the hidden web. Technical Report 2000-36, Computer Science Department, Stanford University, December 2000.
15. Design and Implementation of a High-Performance Distributed Web Crawler Vladislav Shkapenyuk and Torsten Suel
16. Distributing crawling techniques – A survey by Vikas Badgujar, Ashutosh Dixit, A.K.Sharma National conference on emerging trends in computing and Communicating ETTC-07.
17. C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz, "The Harvest Information Discovery and Access System," in Proceedings of the Second International World Wide Web Conference, pp. 763-771, 1994.

AUTHOR'S PROFILE

Mr. Jitendra Kumar Seth received B.Tech. Degree in Computer Science and Engineering from Uttar Pradesh Technical University India, in 2004, and M.Tech Degree in Computer Science and Engineering from Shobhit University, Meerut, India in 2009. He is currently Assistant Professor in Dept. of Information Technology at Ajay Kumar Garg Engg. College, Ghaziabad, India. His research areas are Search Engine, Web Crawlers, Computer Network, java and Web programming, Mobile Computing and Algorithms.

Mr. Ashutosh Dixit is B.Tech, M.Tech in Computer Science. He is currently working as a senior lecturer in YMCA, Faridabad, Hariyana. His research area is web crawler, search engine, computer network and security.

Web Page Prediction Model Based on Clustering Technique

Rajni Pamnani, Pramila Chawan

Department of computer technology,
VJTI University, Mumbai
rajni_as@yahoo.com, pmchawan@vjti.org.in

Abstract -- Users are very often overwhelmed by the huge amount of information and are faced with a big challenge to find the most relevant information in the right time. Prediction systems aim at pruning this information space and directing users toward the items that best meet their needs and interests. Predicting the next request of a user as she visits Web pages has gained importance as Web-based activity increases. Nowadays, Prediction systems are definitely a necessity in the websites not just an auxiliary feature, especially for commercial websites and web sites with large information services. Previously proposed methods for recommendation use data collected over time in order to extract usage patterns. However, these patterns may change over time, because each day new log entries are added to the database and old entries are deleted. Thus, over time it is highly desirable to perform the update of the recommendation model incrementally. In this paper, we propose a new model for modeling and predicting web user sessions which attempt to reduce the online recommendation time while retaining predictive accuracy. Since it is very easy to modify the model, it is updated during the recommendation process.

1 INTRODUCTION

Recommender systems have been used in various applications ranging from predicting the products a customer is likely to buy, movies, music or news that might interest the user and web pages that the user is likely to seek, which is also the focus of this paper. The amount of information available on-line is increasing rapidly with the explosive growth of the World Wide Web and the advent of e-Commerce. Although this surely provides users with more options, at the same time makes it more difficult to find the “right” or “interesting” information from this great pool of information, the problem commonly known as information overload. To address these problems, recommender systems have been introduced. They can be defined as the personalized information technology used to predict a user evaluation of a particular item or more generally as any system that guides users toward interesting or useful objects in a large space of possible options.

Web page recommendation is considered a user modeling or web personalization task. One research area that has recently contributed greatly to this problem is web mining. Most of the systems developed in this field are based on web usage mining which is the process of applying data mining techniques to the discovery of usage patterns from web data. These systems are mainly concerned with analyzing web usage logs, discovering patterns from this data and making recommendations based on the extracted knowledge. One important characteristic of these systems is that unlike traditional recommender systems, which mainly base their decisions on user ratings on different items or other explicit feedbacks provided by the user these techniques discover user preferences from their implicit feedbacks, namely the web pages they have visited. More recently, systems that take advantage of a combination of content, usage and even structure information of the web have been introduced and shown superior results in the web page recommendation problem.

2 MOTIVATION AND RELATED WORK

Today, millions of visitors interact daily with Web sites around the world and massive amounts of data about these interactions are generated. We believe that this information could be very precious in order to understand the user’s behavior. Web Usage Mining is achieved first by reporting visitors traffic information based on Web server log files. For example, if various users repeatedly access the same series of pages, a corresponding series of log entries will appear in the Web log file, and this series can be considered as a Web access pattern.

In the recent years, there has been an increasing number of research works done in Web Usage Mining and their developments. The main motivation of these studies is to get a better understanding of the reactions and motivations of users navigation. Some studies also apply the mining results to improve the design of Web sites, analyze system performances

and network communications or even build adaptive Web sites. We can distinguish three Web Mining approaches that exploit Web logs: association rules (AR), frequent sequences and frequent generalized sequences. Algorithms for the three approaches were developed but few experiments have been done with real Web log data. In this paper, we compare results provided from the three approaches using the same Web log data.

3 CONSIDERATIONS AND DEFINITIONS

With the aim to study Web usage mining, we present in this section our definition of user and navigation.

3.1 User and navigation

A user is identified as a real person or an automated software, such as a Web Crawler (i.e. a spider), accessing files from different servers over WWW. The simplest way to identify users is to consider that one IP address corresponds to one distinct user. A click-stream is defined as a time-ordered list of page views. User's click-stream over the entire Web is called the user session. Whereas, the server session is defined to be the subset of clicks over a particular server by a user, which is also known as a visit. Catledge has studied user page view time over WWW and recommended 25.5 minutes for maximal session length [6]. An episode is defined as a set of sequentially or semantically related clicks.

3.2 Web log files

The easiest way to find information about the users navigation is to explore the Web server logs. The server access log records all requests processed by the server. Server log L is a list of log entries each containing timestamp, host identifier, URL request (including URL stem and query), referrer, agent, etc. Every log entry conforming to the Common Log Format (CLF) contains some of these fields: client IP address or hostname, access time, HTTP request method used, path of the accessed resource on the Web server (identifying the URL), protocol used (HTTP/1.0, HTTP/1.1), status code, number of bytes transmitted, referrer, user-agent, etc. The referrer field gives the URL from which the user has navigated to the requested page. The user agent is the software used to access pages. It can be a spider (ex.: GoogleBot, openbot, scooter, etc.) or a browser (Mozilla, Internet Explorer, Opera, etc.).

4 WEB PAGE PREDICTION MODEL

Following the data cleaning and preprocessing steps, we use similarity metric in the second step for calculating the similarities between all pairs of session. In the third step, the sessions are clustered based on this similarity metric using the graph partitioning algorithm and each cluster is represented by a User Access Stream Tree Model. In order to produce the recommendation set, we first select the cluster and then select the path from the UAST of the cluster that best matches the active user session.

4.1 User Access Stream Tree Model

This model uses both sequence of visiting pages and time spent on each page. In this Model, the sessions are clustered based on this similarity metric using the graph partitioning algorithm and each cluster is represented by a User Access Tree Model. The model we propose has two characteristics:

1. Preservation of whole path of a user session
2. Preservation of time information of visited pages

We generate a click-stream-tree for each cluster. Each user access stream tree has a root node, which is labeled as \null". Each node except the root node of the user access tree consists of three fields: data, count and next node. Data field consists of page number and the normalized time information of that page. Count field registers the number of sessions represented by the portion of the path arriving to that node. Next node links to the next node in the user access stream tree that has the same data field or null if there is any node with the same data field. Each user access stream tree has a data table, which consists of two fields: data field and first node that links to the first node in the user access stream tree that has the data field. The tree for each cluster is constructed by applying the algorithm given in Figure 1.

The children of each node in the click-stream tree is ordered in the count-descending order such that a child node with bigger count is closer to its parent node. The resulting user access stream trees are then used for recommendation.

- 1: Create a root node of a click-stream tree, and label it as "null"
- 2: index \leftarrow 0
- 3: while index \leq number of Sessions in the cluster do
- 4: Active Session \leftarrow index
- 5: m \leftarrow 0
- 6: Current Node \leftarrow root node of the click-stream tree
- 7: while m \leq Active Session length do

```

8: Active Data ← { $p_{t_{index}}^m$ } - { $T_{P_{t_{index}}^m}$ }
9: if there is a Child of Current_Node with the same
data field then
10: Child.count ++
11: Current_Node ← Child
12: else
13: create a child node of the Current_Node
14: Child.data = Active_Data
15: Child.count = 1
16: Current_Node ← Child
17: end if
18: m ++
19: end while
20: index ++
21: end while

```

Figure 1: User Access Stream Tree Algorithm

4.2 Prediction System

The Prediction System is the real time component of the model that selects the best path for predicting the next request of the active user session. The task of the recommendation engine is to compute a *recommendation set* for the current (active) user session, consisting of the objects.

In general there are several design factors that can be taken into account in determining the recommendation set. These factors may include:

- A short-term history depth for the current user representing the portion of the user's activity history that should be considered relevant for the purpose of making recommendations;
- The mechanism used for matching aggregate usage profiles and the active session; and
- A measure of significance for each recommendation (in addition to its prediction value), which may be based on prior domain knowledge or structural characteristics of the site.

Maintaining a history depth is important because most users navigate several paths leading to independent pieces of information within a session. In many cases these *episodes* have a length of no more than two or three references. In such a situation, it may not be appropriate to use references a user made in a previous episode to make recommendations during the current episode. It is possible to capture the user history depth within a sliding window over the current session.

The recommendation engine must be an online process, providing results quickly enough to avoid any perceived delay by the users (beyond what is considered normal for a given Web site and connection speed). There is a trade-off between the prediction accuracy of the next request and the time

spent for recommendation. The speed of the recommendation engine is of great importance in on-line recommendation systems. Thus, we propose the clustering of user sessions in order to reduce the search space and represent each cluster by a click-stream tree. Given the time of the last visited page of the active user session, the model recommends three pages. The most recent visited page of the active user session contains the most important information. The click-stream tree enables us to insert the entire session of a user without any information loss. We not only store the frequent patterns in the tree but also the whole path that a user follows during her session. Besides this, the tree has a compact structure. If a path occurs more than once, only the count of its nodes is incremented.

Based on the construction of the click-stream tree, a path ($p_1, p_2 \dots p_k$), ($T_{p_1}, T_{p_2} \dots T_{p_k}$) occurs in the tree dk .count times, where dk is the data field

formed by merging the page request $P_{t_t}^k$ and corresponding normalized time value $T_{P_{t_t}^k}$ of the path.

4.3 Optimal Path Algorithm

Figure 2 presents the algorithm for finding the path that best matches the active user sessions. For the first two pages of the active user session all clusters are searched to select the best path (line 3). After the second request of the active user top-N clusters that have higher recommendation scores among other clusters are selected (line 29-31) for producing further recommendation sets (line 5). To select the best path we use a backward stepping algorithm. The last visited page and normalized time of that page of the active user session are merged together to build the data field (line 10). We find from the data table of the click-stream tree of a cluster the first node that has the same data field (line 11). We start with that node and go back until the root node (or until the active user session has no more pages to compare) to calculate the similarity of that path to the active user session (line 16-19). We calculate the similarity of the optimal alignment.

To obtain the recommendation score of a path, the similarity is multiplied by the relative frequency of that path, which we define as the count of the path divided by the total number of paths ($S[cl]$) in the tree (line 20). Starting from the first node of the data field and following the next node, the recommendation score is calculated for the paths that contain the data field in the cluster (line 26). The path that has the highest recommendation score is selected as the best path for

generating the recommendation set for that cluster (line 21-24). The first three children nodes of the last node of the best path are used for producing the recommendation set. The pages of these child nodes are recommended to the active user.

Optimal Path Algorithm

```
1:  $t_a \leftarrow$  Active User Session
2: if  $t_a$ .length  $\leq$  2 then
3: Clusters = All Clusters
4: else
5: Clusters = Top-N Clusters
6: end if
7: for  $i = 0$  to NumberOfClusters do
8:  $cl =$  Clusters[ $i$ ]
9:  $Sim[cl] = 0$ 
10:  $d_a \leftarrow \{p_{t_a}^m\} - \{T_{p_{t_a}^m}\}$ 
11: Node  $\leftarrow$  data table[ $cl$ ]( $d_a$ ).first_node
12: path = null
13: while Node  $\neq$  null do
14: path = {path} + {Node.data}
15: Parent Node  $\leftarrow$  Node.Parent
16: while Parent Node  $\neq$  null do
17: path = {path} + {Parent Node.Data}
18: Parent_Node  $\leftarrow$  Parent_Node.Parent
19: end while
20:  $Sim(path) = sim(t_a, path) * Node.count / S[cl]$ 
21: if  $Sim(path) > Sim[cl]$  then
22:  $Sim[cl] \leftarrow Sim(path)$ 
23: BestPath[ $cl$ ]  $\leftarrow$  path
24: end if
25: path = null
26: Node  $\leftarrow$  Node.next_node
27: end while
28: end for
29: if  $t_a$ .length = 2 then
30: Top-N Clusters  $\leftarrow$  N Clusters with highest
    $Sim[cl]$  values
31: end if
```

5 CONCLUSION

One of the goals of Web Usage Mining is to guide the web users to discover useful knowledge and to support them for decision making. In that context, predicting the needs of a web user as he visits web sites has gained importance. The requirement for predicting user needs in order to guide the user in a web site and improve the usability of web site can be addressed by recommending pages to the user that are related to the interest of the user at that time.

This paper proposed a model for discovering and modeling of the user's interest in a single session. This model uses both sequence of visiting pages and time spent on each page. In this Model, the sessions are clustered based on this similarity metric using the graph partitioning algorithm and each cluster is represented by a User Access Tree Model.

REFERENCES

- [1] Renata Ivancsy, István Vajk "Frequent Pattern Mining in Web Log Data",
- [2] Murat Ali Bayir , Ismail Hakki Toroslu "Smart Miner: A New Framework for Mining Large Scale Web Usage Data",
- [3] Mathias G'ery, Hatem Haddad, "Evaluation of Web Usage Mining Approaches for User's Next Request Prediction",
- [4] Wen-Chen Hu, Xuli Zong, "World Wide Web Usage Mining Systems and Technologies",
- [5] Sule Gunduz, M. Tamer Ozsu, "A Web Page Prediction Model Based on ClickStream Tree Representation of User Behavior"
- [6] Sule Gunduz, M. Tamer Ozsu , "Incremental Click-Stream Tree Model: Learning From New Users for Web Page Prediction",
- [7] Natheer Khasawneh, Chien-Chung Chan, "Active User-Based and Ontology-Based Web Log Data Preprocessing for Web Usage Mining"

A Novel Approach to Face Detection using Blob Analysis Technique

D.P.Tripathi¹, S.N.Panigrahy² and Dr.N.P.Rath³

¹Faculty, Roland Institute of Technology, BPUT, Orissa, India

²Faculty, Gandhi Institute of Industrial Technology, BPUT, Orissa, India

³Faculty, Veer Surendra Sai University of Technology, Orissa, India
dpt.tara@gmail.co.in

Abstract: A first step of any face processing system is detecting the locations in images where faces are present. Face detection is one of the challenging problems in the image processing because of variation in scale of the image, location, orientation, pose (frontal, side-view), facial expression, occlusion and lighting condition present which may change the overall appearance of faces in the image. A novel approach to detect faces in a still image is presented here. This method is based on Blob analysis to detect presence of face from a still color image after segmentation with chromatic rules using YCbCr color space, as HSV color components gives lower reliability in complex background and RGB components suffers with changes in the lightning conditions. During blob analysis, the width to height ratio of human face as well as the eccentricity of the blob is taken under consideration. This technique provides good results in single upright frontal face based still color image.

Keywords: Face processing system, Face detection, Image processing, Blobs, YCbCr color space, HSV color space, RGB color space

1. Introduction

With the advancement of new era of Information Technology, more effective and friendly techniques for human-computer interaction are being developed. Moreover with the ever decreasing price of the coupling of computing system with the video Image acquisition techniques imply that computer vision system can be deployed in desktop and embedded systems. This further ignited the research in the field of face processing systems. Many research demonstrations and commercial application have been developed in this way.

Color is known to be a useful cue to extract skin regions, and it is only available in color images. This allows easy face localization of potential facial regions without any consideration of its texture and geometrical properties.

The Chromatic rules are generally defined using RGB, HSV and YCbCr color space. One of the major questions in using skin color in skin detection is how to choose a suitable color space. A wide variety of different color spaces has been applied to the problem of skin color modeling. Monge et al [1] in their review on popular color spaces and their

properties, it is observed that for real world applications and dynamic scenes, color spaces that separate the chrominance and luminance components of color are typically preferable. The main reason for this is that chrominance-dependent components of color increased the robustness to illumination changes in the color images. HSV seems to be a good alternative, but HSV family presents lower reliability when the scenes are complex and they contain similar colors such as wood textures [2].

Moreover during the image acquisition, camera provides RGB image directly, so choice between RGB and YCbCr color comes here. RGB components are subject to the lighting conditions thus the face detection may fail if the lighting condition changes. So YCbCr color space is used here for skin color detection.

From the detected skin region the region of interest have been found and blob are been detected. It is found that the human faces are generally oval in shape and they are satisfying a particular range of eccentricity and width-height ratio. Here in this paper we have presented the face detection using these unique features to detect presence of face in a single upright frontal still color image.

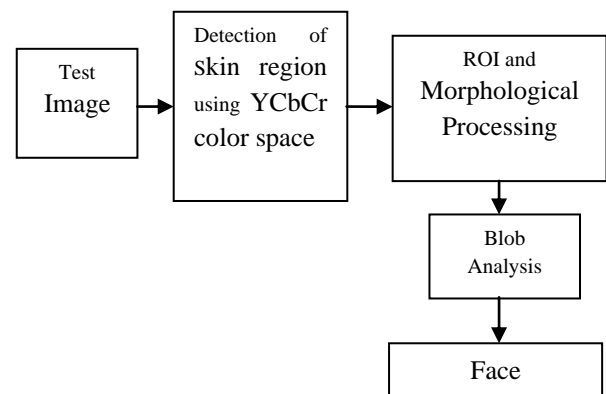


Figure 1. The proposed Face Detection System

2. Face Detection Algorithm

The proposed face detection system shown in Figure-1 consists of following categories; Detection of Skin-region using YCbCr color space. Extraction of Region of Interests (ROI), Classification of face region using blob analysis technique.

2.1. Detection of Skin-region using YCbCr color space

Color is a prominent feature of human faces. Using skin color as a primitive feature for detecting face regions has several advantages. In particular, processing color is much faster than processing other facial features. Furthermore, color information is invariant against scaling, rotations, translation and skewing. However, even under a fixed ambient lighting, people have different skin color appearance. In order to effectively exploit skin color for face detection, we need to find a feature space, in which human skin colors cluster tightly together and reside remotely to background colors.

Rowley et al [3], Hunke [4], Esteki [5] and Hsu et al [6] have stated different techniques for locating skin color regions in the still color images.

While the input color image is typically in the RGB format, these techniques usually use color components in the color space, such as the HSV or YIQ formats. That is because RGB components are subject to the lighting conditions thus the face detection may fail if the lighting condition changes. Peer et al [7] designed a set of rules to cluster skin regions in RGB color space. Moreover since the reliability of HSV components become poorer with the increase in background complexity [2]. Thus this project used YCbCr components and as it is one of the existing Matlab functions thus would save the computation time. Rahman et al [8] and Sabottka et al [9] also observed that the varying intensity value of Y (Luminance) does not alter the skin color



Figure 2. The Original Test Image

distribution in the Cb-Cr subspace. In the YCbCr color space, the luminance information is contained in Y component and the chrominance information is in Cb and Cr.

Therefore, the luminance information can be easily de-embedded. The RGB components were converted to the YCbCr components using the following formula.

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.169R - 0.332G + 0.500B \\ Cr &= 0.500R - 0.419G - 0.081B \end{aligned} \quad (1)$$

To design a skin detector which is used as the basis for skin-like region segmentation, skin samples from 60 training faces (male as well as female) are used.

These samples are cropped into same size (15x15 pixels) and are filtered using a low-pass filter to reduce the effect of noise in the samples. Cb and Cr components of each sample are calculated. Then mean and covariance values are obtained from these Cb and Cr values.

$$\begin{aligned} \text{Mean, } m &= E(x) \text{ where } x = (Cr, Cb)^T \\ \text{Covariance, } C &= E((x-m)(x-m)^T) \end{aligned} \quad (2)$$

The original Test image which is in RGB format is converted into YCbCr according to relation (1). Then the likelihood of skin for each pixel is computed using the relation given below [10];

$$\text{Likelihood} = P(Cr, Cb) = \exp[-0.5(x-m)^T C^{-1}(x-m)] \quad (3)$$

Since people with different skin have different likelihood, so to improve the reliability of skin-detector adaptive thresholding process is used to achieve optimal threshold value for each run.

This adaptive thresholding is based on the observation that stepping the threshold value down may increase the segmented region. The threshold value at which the minimum increase in region size is observed while stepping down the threshold value is the optimal threshold. Here we have



Figure 3. The Skin likelihood Image

decremented the threshold value from 0.67 to 0.05 in steps of 0.1. Using the above procedure skin-like regions (Figure-3) are detected from the original image (Figure-2).

Now the next section will describe the steps for processing the image obtained in Figure (3) in order to obtain the Region of Interest (ROI) of possible skin areas in a test image.

2.2. Extraction of Region of Interests (ROIs)

The importance of region of interest (ROI) is to highlight the area for processing and differentiate it from the rest of the

image. All the pixel values which have likelihood values higher than optimal threshold value are set to 1 and the rest of pixels are set to 0. Thus resulting a binary image.

After getting a binary image, morphological operations [11] such as filling, erosion and dilation are applied to separate skin areas which are loosely connected to obtain the region of interest shown in figure(4). Morphological erosion and dilation are applied by using structural element disk of size 8. Then the images are extracted from the ROI shown in figure(5).



Figure 4. The Region of Interests (ROIs)



Figure 5. Extracted Images according to Defined ROIs

2.3. Classification using Blob Analysis Technique

Human faces are generally oval in shape. The analysis of the characteristics of this oval shape provides the cue for the presence of face.

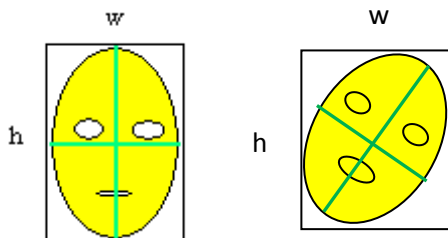


Figure 6. Oval shape characteristics of human face

The oval shape geometry shown in figure(6) provided two shorts of important parameter i.e the boundingbox ratio and eccentricity to examine and classify the shape of detected skin region as a 'face' or 'non-face'.

Each detected skin region can be made bounded by a box. The boundingbox ratio can be defined as the ratio of the width(w) to height(h) of the box encompass the region. After experimenting with 35 numbers of face regions and 55 numbers of non-face regions, it has been found that the ratio generally lies between 1.0 to 0.4. Face regions are generally having ratio less than 1 but value less than 0.4 indicates the presence of non-face region.

Again the oval nature of the face provides another parametet i.e eccentricity which is the ratio of the length of minor axis to the major axis. These axis are shown in the figure(6) with green color line segment. Eccentricity property is more sensitive to the oval shape geometry for characterization. On experimenting again with the same numbers of face and non-face regions. It has found that the eccentricity value lies between 0.3 to 0.9 provides good classification between face and non-face.



Figure 7. The detected face region

On applying both the rules simultaneously on the extracted skin regions shown in figure (5), successfully the face region is detected shown in figure (7).

3. Experimental Results

Unlike in face recognition where the classes to be discriminated are different faces, in face detection the two classes are "face area" and "non-face area".

This face detection system was implemented using MATLAB 7.0 on a Intel Core2Duo processor with 2GB RAM.

The proposed Blob analysis technique along with YCbCr adaptive segmentation technique was experimented with 65 number of color images having 135 faces and 210 non-faces.

The performance of the system was evaluated with the parameter i.e successful detection rate(SDR), which has been defined as the percentage ratio of the successful detected faces to the total number of faces.

$$SDR = \frac{\text{Successfully detected faces}}{\text{Total number of faces}} \times 100 \% \quad (4)$$

The proposed face detection system achieved a good detection rate of 95.6% by detecting successfully 129 faces out of 135.



(a)

(b)



(c)



(d)

Figure 8. Detected Face images

4. Conclusion

This paper presents in detail the application of YCbCr color space and Blob analysis technique with necessary image processing procedures to detect the presence of human faces in color images.

In face detection system because of the different illumination conditions and variety of positions of the face in the image, detecting a face becomes a tedious task.

The problem of illumination can be more accurately solved using these YCbCr color space by carefully choosing threshold range for skin color detection.

In future work the problem relating to occlusion of face in an image can be removed by refine use of Morphological operations on the extracted blob regions.

Moreover, it can become an efficient pre-processing step of a face recognition system, as the falsely detected faces can be easily eliminated out using correspondence with the database of already known faces as used in recognition system.

References

- [1] Jesus Ignacio Buendo Monge "Hand Gesture Recognition for Human-Robot Interaction". Master of Science thesis KTH. Sweden 2006.
- [2] Jay P. Kapur, EE499 Capstone Design Project Spring 1997, University of Washington, Department of Electrical Engineering.
- [3] H. Martin Hunke, Locating and tracking of Human Faces with neural Network, Master's Thesis, University of Karlsruhe, 1994.
- [4] Henry A. Rowley, S. Baluja, and Takco Kanade, "Neural Network Based face Detection", IEEE trans. On Pattern Analysis and Machine Intelligence, 20(1), pp. 23-38, 1998.
- [5] Hadi Esteki, "Face Detection and Recognition in Indoor Environment", Master of Science Thesis, Stockholm, Sweden, 2007
- [6] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, Anil K Jain, "Face Detection in Color Images", IEEE transactions on Pattern Analysis and Machine Intelligence, vol 24, No.5, pp. 696-706, May 2002.
- [7] P. Peer, J. Kovac, F. Solina, "Human Skin Colour Clustering for Face Detection", EUROCON1993, Ljubljana, Slovenia, pp. 144-148, September 2003.
- [8] Nusirwan Anwar bin Abdul Rahman, Kit Chong Wei and John See, "RGB-H-CbCr Skin Colour Model for Human Face Detection" Faculty of Information Technology, Multimedia University.
- [9] K. Sabottka and I. Pitas, "Segmentation and Tracking of Faces in Color Images", AFGR'96, Killington, Vermont, pp. 236-241, 1996.
- [10] Weng Foo Henry Chang and Ulises Robles-Mellin "Face Detection", EE368 Final project report-spring-2000.

- [11] Gonzales R. C., Woods R.E., “Digital Image Processing”, Prentice Hall, pp.524-526, 2001.

Author Biography



Durga P. Tripathi obtained a bachelor's degree in Electronics & Telecommunication Engineering in the year 2002 from Fakir Mohan University, India and Completed his M.Tech in the year 2009 in Communication System Engineering in Veer Surendra Sai University Of Technology, Orissa, India. At present working as a lecturer in Department of Electronics and Telecommunication ,RIT, Berhampur, Orissa.

Survey on Multimedia Operating Systems

P. DharanyaDevi, S. Poonguzhali, T. Sathiya, G.Yamini, P. Sujatha and V. Narasimhulu

Department of Computer Science, Pondicherry Central University,
Pondicherry - 605014, India.
{spothula, narasimhavasi}@gmail.com

Abstract: Real-time applications such as multimedia audio and video are increasingly populating the workstation desktop. A growing number of multimedia applications are available, ranging from video games and movie players, to sophisticated distributed simulation and virtual reality environment. Multimedia is an increasingly important part of the mix of applications that users run on personal computers and workstations. Research in operating system support for multimedia has traditionally been evaluated using metrics such as fairness, the ability to permit applications to meet real-time deadlines, and run-time efficiency. In addition, the support for real-time applications is integrated with the support for conventional computations. This poster deals with the survey on multimedia operating systems, its process scheduling, disk management, file management and device management techniques.

Keywords: Multimedia Operating Systems, CPU Scheduling, Memory Management, Device Management, File Management.

1. Introduction

Multimedia data demands strict time constraints for processing. In any multimedia application, we may have several processes running dependently on one another [12]. For example, one process may generate video frames for an X-window process while another process generates an audio stream for an attached speaker system. These two processes must execute in parallel for the application to be of any worth. In other words, the processes require relative progress to one another. It is of no use to begin executing the audio process once the video is half finished. Certain media processes may require absolute time progress as well. For example, the video application should process frames at a constant rate with respect to world time. If steady absolute progress is not enforced, one would observe random stopping and starting of the video. If relative progress is not enforced, cooperating processes such as the audio and video application mentioned earlier will not function properly.

Multimedia can be classified as live-data applications or stored-data applications. Live-data is much harder to process effectively because there can be little or no data buffering to ensure consistent output. For live-data, displaying audio and video as it happens reduces the amount of slack time allowed for computation and resource scheduling. Live-data is simply more demanding in its temporal deadlines than stored multimedia data. Stored-data can be retrieved in bulk well in advance of output deadlines. This ensures data will be available most of the time for processing when required [27].

Personal computers running Windows XP, MacOS X, and Linux are capable of performing a variety of multimedia tasks accurately recognizing continuous speech, encoding

captured television signals and storing them on disk, acting as professional-quality electronic musical instruments, and rendering convincing virtual worlds all in real time[5]. Furthermore, personal computers costing less than \$1000 are capable of performing several of these tasks at once if the operating system manages resources well [21] [22]. The increasing pervasiveness of multimedia applications, and problems supporting them on traditional systems, has motivated many research papers over the past decade.

In this paper, in section 2 we illustrate some features of multimedia OS, section 3 describes CPU scheduling techniques, section 4 describes memory management, section 5 describes device management, section 6 describes file management, section 7 describes disk scheduling algorithms and finally section 8 describes the conclusion of the paper.

2. Multimedia Requirements

A general-purpose operating system (GPOS) for a personal computer or workstation must provide fast response time for interactive applications, high throughput for batch applications, and some amount of fairness between applications [10] [13]. Although there is tension between these requirements the lack of meaningful changes to the design of time-sharing schedulers in recent years indicates that they are working well enough. The goal of a hard real-time system is similarly unambiguous: all hard deadlines must be met. The standard engineering practice for building these systems is to statically determine resource requirements and schedulability, as well as over-provisioning resources as a hedge against unforeseen situations.

We have identified four basic requirements that the "ideal" multimedia operating system should meet [23]. Although it is unlikely that any single system or scheduling policy will be able to meet all of these requirements for all types of applications, the requirements are important because they describe the space within which multimedia systems are designed. A particular set of prioritizations among the requirements will result in a specific set of tradeoffs; these tradeoffs will constrain the design of the user interface and the application programming model.

1. *Meet the scheduling requirements of coexisting, independently written, possibly misbehaving soft real time applications:* The CPU requirements of a real-time application are often specified in terms of an *amount* and *period*; here the application must receive the amount of CPU time during each period of time. No matter how scheduling requirements are specified, the scheduler must be able to

meet them without the benefit of global coordination among application developers multimedia operating systems are *open systems* in the sense that applications are written independently [6].

2. *Minimize development effort by providing abstractions and guarantees that are a good match for applications requirements:* In the past, personal computers were dedicated to a single application at a time. Developers did not need to interact much with OS resource allocation policies. This is no longer the case. For example, it is possible to listen to music while playing a game, burn a CD while watching a movie, or encode video from a capture card while using speech recognition software. Therefore, an important role of the designers of soft real-time systems is to make it as easy as possible for developers to create applications that gracefully share machine resources with other applications [6].

3. *Provide a consistent, intuitive user interface:* Users should be able to easily express their preferences to the system and the system should behave predictably in response to user actions. Also, it should give the user (or software operating on the user's behalf) feedback about the resource usage of existing applications and, when applicable, the likely effects of future actions [6].

4. *Run a mix of applications that maximizes overall value:* Unlike hard real-time systems, PCs and workstations cannot overprovision the CPU resource; demanding multimedia applications tend to use all available cycles. During overload the multimedia OS should run a mix of applications that maximizes overall value. This is the "holy grail" of resource management and is probably impossible in practice since value is a subjective measure of the utility of an application, running at a particular time, to a particular user. Still, this requirement is a useful one since it provides a basis for evaluating different systems [6].

3. CPU Scheduling

3.1 Fixed-Time Allocation

By giving real-time computation, a higher priority than system and other user processes, it limits the utilization of the system and artificially constrains the range of behavior the system can provide. Priority real-time execution can cause system services to lock up, and the user can lose control over the machine.

Real-time processes are allocated the CPU first. They are allowed to execute for a fixed amount of time. If some processes are not able to meet their deadline within the fixed amount of time allocated, the process is notified it will miss its deadline. The process may abort or continue executing, depending on the application [8]. Conventional processes are then allocated a fixed amount of CPU time. The cycle continues, alternating between conventional and real-time execution.

The amount of time allocated for real-time and conventional computation depends on the workload ratio. A good solution when the real-time deadlines can be met within the fixed time allocated. In this case the scheduler provides adequate service for all classes of computation.

3.2 Rate-Based Priority Scheduling

It is also called rate-based adjustable priority scheduling (RAP). The algorithm makes three assumptions about the real-time processes it schedules [8].

1. RAP does not assume a priori knowledge of resource requirements by MM applications.
2. RAP assumes multimedia applications can tolerate occasional delays in execution.
3. RAP assumes MM applications are adaptive in nature and can gracefully adapt to resource overloads by modifying their behavior to reduce their resource requirements.

At the beginning of execution, an application specifies a desired average rate of execution and a time interval over which the average rate of execution will be measured. RAP implements an admission control scheme that calculates the available CPU capacity and compares it to the requested execution rate. If an acceptable execution rate can be allocated, then the process is placed in the set of runnable processes. The queue of real-time processes is organized on a priority basis. Each process priority is based on the requested rate of execution. It is not clear how the priority relates to the rate of execution.

Once a process is admitted to the set of runnable processes, the scheduler allocates the CPU using a priority-based scheduler and a rate regulator. The rate regulator ensures a process which was promised an average execution rate R does not execute more than R times a second and executes roughly once every $T=1/R$ time interval. After a process executes for the duration of one averaging interval, feedback is provided back to the application about the observed rate of progress [2]. The quality-of-service manager assumed to be implemented in the application reacts accordingly. It may increase or decrease its desired rate of execution. RAP also has a mechanism that monitors CPU capacity. If the CPU is over or under-utilized, it can communicate with application level processes to decrease or increase its resource demands by a fraction of its current demand, respectively.

This algorithm provides a good basis for future work in system and application layer cooperation. As opposed to some of the other scheduling techniques described which were entirely system or application based, this scheduler is effectively implemented at both the application and operating system level. Communication and cooperation of the two levels help establish a fair and adaptable scheduling discipline.

3.3 Earliest Deadline First

When the scheduler is in real-time mode, the processes are scheduled in an earliest deadline first scheme. Conventional processes are allocated in a round-robin discipline. The Earliest Deadline first scheduling is theoretically optimal under certain assumptions. Soft real time OS uses EDF as an internal scheduler. Only a few systems such as Rialto and SMART expose deadline-based scheduling abstractions to application programmers. Both systems couple deadline-

based scheduling with an admission test and call the resulting abstraction a time constraint.

Time constraints present a fairly difficult programming model because they require fine-grained effort: the developer must decide which pieces of code to execute within the context of a time constraint in addition to providing the deadline and an estimate of the required processing time. Applications must also be prepared to skip part of their processing if the admission test fails. Once a time constraint is accepted, Rialto guarantees the application that it will receive the required CPU time. SMART, on the other hand, will sometimes deliver an up call to applications informing them that a deadline previously thought to be feasible has become infeasible, forcing the program to take appropriate action [8].

3.4 Feedback-Based Scheduling

Multimedia OS need to work in situations where total load is difficult to predict and execution times of individual applications vary considerably. To address these problems new approaches based on feedback control have been developed. Feedback control concepts can be applied at admission control and/or as the scheduling algorithm itself. In the FC-EDF work [1] a feedback controller is used to dynamically adjust CPU utilization in such a manner as to meet a specific set point stated as a deadline miss percentage. FC-EDF is not designed to prevent individual applications from missing their deadlines; rather, it aims for high utilization and low overall deadline miss ratio.

SWiFT uses a feedback mechanism to estimate the amount of CPU time to reserve for applications that are structured as pipelines. The scheduler monitors the status of buffer queues between stages of the pipeline; it attempts to keep queues half full by adjusting the amount of processor time that each stage receives. Both SWiFT and FC-EDF have the advantage of not requiring estimates of the amount of processing time that applications will need. Both systems require periodic monitoring of the metric that the feedback controller acts on.

3.5 Hierarchical Scheduling

Hierarchical schedulers generalize the traditional role of schedulers by allowing them to allocate CPU time to other schedulers. The *root* scheduler gives CPU time to a scheduler below it in the hierarchy and so on until a leaf of the scheduling tree. The scheduling hierarchy may either be fixed at system build time or dynamically constructed at run time. *CPU inheritance scheduling* [3] probably represents an endpoint on the static vs. dynamic axis: it allows arbitrary user-level threads to act as schedulers by *donating* the CPU to other threads.

Hierarchical scheduling has two important properties. First, it permits multiple programming models to be supported simultaneously, potentially enabling support for applications with diverse requirements. Second, it allows properties that schedulers usually provide to threads to be recursively applied to groups of threads [7].

The comparison of fixed-time allocation, rate-based priority and hierarchical scheduling is given in Table 1.

Table 1. Comparison of scheduling algorithms

Features	Fixed – Time Allocation	Rate-Based Priority Scheduling	Hierarchical Scheduling
Nature of Allocation	Static	Dynamic	Dynamic
Priori knowledge of needed resources	Required	Required	Not Required
Scheduling Mechanism	Earliest Deadline First, Round Robin	Priority Based	Not specific
Admission Control Used	No	Yes	No
Monitors Used	No	Rate Regulator, QoS Manager	No
Efficiency	Average	Good	Best
Support Hard and Soft real-time	No	No	Yes

3.6 CPU Schedulers

In the following subsections, we describe in more detail two distinct schedulers.

3.6.1 Rialto scheduler

The scheduler of the Rialto OS is based on three fundamental abstractions:

- *Activities* are typically an executing program or application that comprises multiple threads of control. Resources are allocated to activities and their usage is charged to activities.
- *CPU reservations* are made by activities and are requested in the form: “reserve x units of time out of every Y units for activity A ”. Basically, period length and reservations for each period can be of arbitrary length.
- *Time constraints* are dynamic requests from threads to the scheduler to run a certain code segment within a specified start time and deadline to completion.

The scheduling decision [6], i.e. which threads to activate next, is based on a pre-computed scheduling graph. Each time a request for CPU reservation is issued, this scheduling graph is recomputed. In this scheduling graph, each node represents an activity with a CPU reservation, specified as time interval and period, or represents free computation time.

For each base period, i.e. the lowest common denominator of periods from all CPU reservations, the scheduler traverses the tree in a depth-first manner, but back tracks always to the root after visiting a leaf in the tree. Each node, i.e. activity that is crossed during the traversal, is scheduled for the specified amount of time.

The execution time associated with the schedule graph is fixed. Free execution times are available for non-time-critical tasks. This fixed schedule graph keeps the number of context switches low and keeps the scheduling algorithm scalable. If threads request time constraints, the scheduler analyzes their feasibility with the so-called *time interval assignment* data structure. This data structure is based on the

knowledge represented in the schedule graph and checks whether enough free computation time is available between start time and deadline (including the already reserved time in the CPU reserve).

Threads are not allowed to define time constraints when they might block—except for short blocking intervals for synchronization or I/O. When during the course of a scheduling graph traversal an interval assignment record for the current time is encountered, a thread with an active time constraint is selected according to EDF [6]. Otherwise, threads of an activity are scheduled according to round-robin. Free time for non-time-critical tasks is also distributed according to round-robin. If threads with time constraints block on a synchronization event, the thread priority (and its reservations) is passed to the holding thread.

3.6.2 SMART scheduler

The SMART scheduler [6] is designed for multimedia applications and is implemented in Solaris 2.5.1. The main idea of SMART is to differentiate between *importance* to determine the overall resource allocation for each task and *urgency* to determine when each task is given its allocation. Importance is valid for real-time and conventional tasks and is specified in the system by a tuple of priority and biased virtual finishing time.

Here, the virtual finishing time [4], as known from fair-queuing schemes, is extended with a bias, which is a bounded offset measuring the ability of conventional tasks to tolerate longer and more varied service delays. Application developers can specify time constraints, i.e. deadlines and execution times, for a particular block of code, and they can use the system notification.

The system notification is an up call that informs the application that a deadline cannot be met and allows the application to adapt to the situation. Applications can query the scheduler for availability, which is an estimate of processor time consumption of an application relative to its processor allocation. Users of applications can specify priority and share to bias the allocation of resources for the different applications.

The SMART scheduler separates importance and urgency considerations. First, it identifies all tasks that are important enough to execute and collects them in a candidate set. Afterwards, it orders the candidate set according to urgency consideration.

In more detail, the scheduler works as follows [26]: if the tasks with the highest value-tuple are a conventional task, schedule it. The highest value-tuple is either determined by the highest priority or for equal priorities by the earliest biased virtual finishing time. If the task with the highest value-tuple is a real-time task, it creates a candidate set of all real-time tasks that have a higher value tuple than the highest conventional task. The candidate set is scheduled according to the so-called best-effort real-time scheduling algorithm.

Basically, this algorithm finds the task with the earliest deadline that can be executed without violating deadlines of tasks with higher value-tuples. SMART notifies applications if their computation cannot be completed before its deadline. This enables applications to implement downscaling. There is no admission control implemented in SMART. Thus, SMART can only enforce real-time behavior in underload situations.

3.6.3 EScheduler

EScheduler, an energy-efficient soft real-time CPU scheduler [6] for multimedia applications running on a mobile device. *EScheduler* seeks to minimize the total energy consumed by the device while meeting multimedia timing requirements. To achieve this goal, *EScheduler* integrates *dynamic voltage scaling* into the traditional soft real-time CPU scheduling: It decides *at what CPU speed* to execute applications in addition to when to execute what applications. *EScheduler* makes these scheduling decisions based on the probability distribution of cycle demand of multimedia applications and obtains their demand distribution via online profiling.

(a) Advantages of EScheduler

1. Scheduling is stable. This stability implies the feasibility to perform our proposed energy-efficient scheduling with low overhead.
2. *EScheduler* delivers soft performance guarantees to these codecs by bounding their deadline miss ratio under the application-specific performance requirements.
3. *EScheduler* reduces the total energy of the laptop by 14.4 percent; to 37.2 percent; relative to the scheduling algorithm without voltage scaling and by 2 percent; to 10.5 percent; relative to voltage scaling algorithms without considering the demand distribution.
4. *EScheduler* saves energy by 2 percent; to 5 percent; by explicitly considering the discrete CPU speeds and the corresponding total power of the whole laptop, rather than assuming continuous speeds and cubic speed-power relationship.

Table 2. Comparison of CPU schedulers

The comparison of CPU schedulers in terms of features is given in Table2.

4. Memory Management

Techniques such as demand-paging and memory-mapped files have been successfully used in commodity OS. However, these techniques fail to support multimedia

Features	SMART	Rialto	EScheduler
Platform	Solaris 2.5.1	Not specific	Mobile device
Time constraints	Dynamic	Dynamic	Static
Scheduling mechanism Based on	Virtual finishing time	Recomputed graph	Dynamic voltage scaling
Admission Control	No	No	No
Support for Hard real time application	No	No	Yes
Scheduling Algorithm	Best-effort real-time	Hierarchical	Proportional share

applications, because they introduce unpredictable memory

access times, cause poor resource utilization, and reduce performance. In the following subsections, we present new approaches for memory allocation and utilization, data replacement, and prefetching using application-specific knowledge to solve these problems. Furthermore, we give a brief description of the UVM Virtual Memory System that replaces the traditional virtual memory system in NetBSD 1.4. [5]

4.1 Memory Allocation

Usually, upon process creation, a virtual address space is allocated which *contains* the data of the process. Physical memory [20] is then allocated and assigned to a process and then mapped into the virtual address space of the process according to available resources and a global or local allocation scheme. This approach is also called *user-centered allocation* [6]. Each process has its own share of the resources. However, traditional memory allocation on a per client (process) basis suffers from a linear increase of required memory with the number of processes. In order to better utilize the available memory, several systems use so-called *data-centered allocation* where memory is allocated to data objects rather than to a single process. Thus, the data is seen as a resource principal. This enables more cost-effective data-sharing techniques [16] [18]:

(1) **Batching** starts the video transmission when several clients request the same movie and allows several clients to share the same data stream;

(2) **Buffering** (or *bridging*) caches data between consecutive clients omitting new disk requests for the same data.

(3) **Stream merging** (or *adaptive piggy-backing*) displays the same video clip at different speeds to allow clients to catch up with each other and then share the same stream.

(4) **Content insertion** is a variation of stream merging, but rather than adjusting the display rate, new content, e.g. commercials, is inserted to align the consecutive playouts temporally;

(5) **Periodic services** (or *enhanced pay-per-view*) assign each clip a retrieval period where several clients can start at the beginning of each period to view the same movie and to share resources.

These data-sharing techniques are used in several systems. All buffers are shared among the clients watching the same movie and work like a sliding window on the continuous data [14]. When the first client has consumed nearly all the data in the buffer, it starts to refresh the oldest buffers with new data. Periodic services are used in pyramid broadcasting. The data is split in partitions of growing size, because the consumption rate of one partition is assumed to be lower than the downloading rate of the subsequent partition. Each partition is then broadcast in short intervals on separate channels.

A client does not send a request to the server, but instead it tunes into the channel transmitting the required data. The data is cached on the receiver side, and during the playout of a partition, the next partition is downloaded. However, to avoid very large partitions at the end of a movie and thus to reduce the client buffer requirement, the partitioning is

changed such that not every partition increases in size, but only each n th partition. Performance evaluations show that the data-centered allocation schemes scale much better with the numbers of users compared to user-centered allocation. The total buffer space required is reduced, and the average response time is minimized by using a small partition size at the beginning of a movie.

The **memory reservation per storage device** mechanism allocates a fixed, small number of memory buffers per storage device in a server-push VoD server using a cycle based scheduler. [19] In the simplest case, only two buffers of identical size are allocated per storage device. These buffers work co-operatively, and during each cycle, the buffers change task as data is received from disk. That is, data from one process is read into the first buffer, and when all the data is loaded into the buffer, the system starts to transmit the information to the client. At the same time, the disk starts to load data from the next client into the other buffer. In this way, the buffers change task from receiving disk data to transmitting data to the network until all clients are served. The admission control adjusts the number of concurrent users to prevent data loss when the buffers switch and ensures the maintenance of all client services [24].

4.2 Data Replacement

When there is need for more buffer space, and there are no available buffers, a buffer has to be replaced. How to best choose which buffer to replace depends on the application. However, due to the high data consumption rate in multimedia applications, data is often replaced before it might be reused. The gain of using a complex page replacement algorithm might be wasted and a traditional algorithm as. Nevertheless, in some multimedia applications where data often might be reused, proper replacement algorithms may increase performance. The *distance*, the *generalized interval caching* and the SHR schemes, all replace buffers after the distance between consecutive clients playing back the same data and the amount of available buffers [6].

Usually, data replacement is handled by the OS kernel where most applications use the same mechanism. Thus, the OS has full control, but the used mechanism is often tuned to best overall performance and does not support application specific requirements.

Self-paging has been introduced as a technique to provide QoS to multimedia applications. The basic idea of self-paging is to “require every application to deal with all its own memory faults using its own concrete resources”. All paging operations are removed from the kernel where the kernel is only responsible for dispatching fault notifications. This gives the application flexibility and control, which might be needed in multimedia systems, at the cost of maintaining its own virtual memory operations. However, a major problem of self-paging is to optimize the global system performance. Allocating resources directly to applications gives them more control, but that means optimizations for global performance improvement are not directly achieved.

4.3 Prefetching

The poor performance of demand-paging is due to the low disk access speeds. Therefore, prefetching data from disk to memory is better suited to support continuous playback of time-dependent data types. Prefetching is a mechanism to preload data from slow, high-latency storage devices such as disks to fast, low-latency storage like main memory [6]. This reduces the response time of a data read request dramatically and increases the disk I/O bandwidth.

Prefetching mechanisms in multimedia systems can take advantage of the sequential characteristics of multimedia presentations. For example, a read-ahead mechanism retrieves data before it is requested if the system determines that the accesses are sequential. The utilization of buffers and disk is optimized by prefetching all the shortest database queries maximizing the number of processes that can be activated once the running process is finished. Assuming a linear playout of the continuous data stream, the data needed in the next period (determined by a tradeoff between the maximum concurrent streams and the initial delay) is prefetched into a shared buffer [15].

In addition to the above-mentioned prefetching mechanisms designed for multimedia applications, more general purpose facilities for retrieving data in advance are designed which also could be used for certain multimedia applications.

The **informed prefetching** and caching strategy preloads a certain amount of data where the buffers are allocated / deallocated according to a global max-min valuation. This mechanism is further developed. Where the automatic hint generation, based on speculative pre-executions using mid-execution process states, is used to prefetch data for possible future read requests.

Moreover, the **dependent-based prefetching** captures the access patterns of linked data structures. A prefetch engine runs in parallel with the original program using these patterns to predict future data references. Finally, an analytic approach to file prefetching is described. During the execution of a process a semantic data structure is built showing the file accesses. When a program is re-executed, the saved access trees are compared against the current access tree of the activity, and if a similarity is found, the stored tree is used to preload files.

Obviously, knowledge (or estimations) about application behavior might be used for both replacement and prefetching. A multimedia object is replaced and prefetched according to its relevance value computed according to the presentation point/modus of the data playout.

4.4 Cache Management

All real-time applications rely on predictable scheduling, but the memory cache design makes it hard to forecast and schedule the processor time. Furthermore, memory bandwidth and the general OS performance have not increased at the same rate as CPU performance. Benchmarked performance can be improved by enlarging and speeding up static RAM-based cache memory, but the large amount of multimedia data that has to be handled by CPU and memory system will likely decrease cache hit ratios [6]. If two processes use the same cache lines and are executed concurrently, there will not only be an increase in

context switch overheads, but also a cache-interference cost that is more difficult to predict. Thus, the system performance may be dominated by slower main memory and I/O accesses. Furthermore, the busier a system is, the more likely it is that involuntary context switches occur; longer run queues must be searched by the scheduler, etc. flushing the caches even more frequently.

UVM virtual memory system: The UVM Virtual Memory System replaces the virtual memory object, fault handling, and pager of the BSD virtual memory system; and retains only the machine dependent/independent layering and mapping structures [6]. For example, the memory mapping is redesigned to increase efficiency and security; and the map entry fragmentation is reduced by memory wiring.

In BSD, the memory object structure is a stand-alone abstraction and under control of the virtual memory system. In UVM, the memory object structure is considered as a secondary structure designed to be embedded with a handle for memory mapping resulting in better efficiency, more flexibility, and less conflicts with external kernel subsystems. The new copy-on-write mechanism avoids unnecessary page allocations and data copying, and grouping or clustering the allocation and use of resources improves performance. Finally, a virtual memory-based data movement mechanism is introduced which allows data sharing with other subsystems, i.e. when combined with the I/O or IPC systems; it can reduce the data copying overhead in the kernel.

4.5 Management of other resources

This section takes a brief look at management aspects of OS resources.

4.5.1 Speed Improvements in Memory Access

The term dynamic RAM (DRAM), coined to indicate that any random access in memory takes the same amount of time, is slightly misleading. Most modern DRAMs provide special capabilities that make it possible to perform some accesses faster than others [5]. For example, consecutive accesses to the same row in a page mode memory are faster than random accesses, and consecutive accesses that hit different memory banks in a multi-bank system allow concurrency and are thus faster than accesses that hit the same bank.

The key point is that the order of the requests strongly affects the performance of the memory devices. The most common method to reduce latency is to increase the cache line size, i.e. using the memory bandwidth to fill several cache locations at the same time for each access.

However, if the stream has a non-unit-stride (stride is the distance between successive stream elements in memory), i.e. the presentation of successive data elements does not follow each other in memory, and the cache will load data which will not be used. Thus, lengthening the cache line size increases the effective bandwidth of unit-stride streams, but decreases the cache hit rate for non-streamed accesses. Another way of improving memory bandwidth in memory-cache data transfers for streamed access patterns

4.5.2 Multimedia mbuf

The *multimedia mbuf* (mmbuf) is specially designed for disk-to-network data transfers [6]. It provides a zero-copy data path for networked multimedia applications by unifying the buffering structure in file I/O and network I/O. This buffer system looks like a collection of clustered mbufs that can be dynamically allocated and chained. The mmbuf header includes references to mbuf header and buffer cache header. By manipulating the mmbuf header, the mmbuf can be transformed either into a traditional buffer, that a file system and a disk driver can handle, or an mbuf, which the network protocols and network drivers can understand.

A new interface is provided to retrieve and send data, which coexist with the old file system interface. The old buffer cache is bypassed by reading data from a file into an mmbuf chain. Both synchronous (blocking) and asynchronous (non-blocking) operations are supported and read and send requests for multiple streams can be bunched together in a single call minimizing system call overhead. At setup time, each stream allocates a ring of buffers, each of which is an mmbuf chain.

5. Device Management

A device management system for supporting applications that reside on a multimedia client, the applications interacting with a plurality of stream devices associated with the multimedia client, comprising: a stream manager being configured to identify the plurality of stream devices and store a device identifier for each of said stream devices [6].

The first application being operative to initiate communication between a first stream device and said first application by sending a device identifier to said stream manager, said device identifier indicative of said first stream device; and said stream manager being operative, in response to receiving a device identifier from said first application, to stream data between said first application and said first stream device [6].

A stream device management system is provided for supporting applications that access a variety of stream devices associated with a conventional set-top box. More specifically, the stream device management system includes a stream manager configured to identify a plurality of stream devices and to store a device identifier for each of these stream devices, and a shared memory for storing stream data associated with each of the stream devices.

To initiate communication with a first stream device, a first application sends a device identifier indicative of the first stream device to the stream manager. In response to receiving the device identifier, the stream manager communicates an address for the shared memory associated with the first stream device to the first application. Lastly, the application uses this address to access the stream data.

6. File System

The file system plays a major role in every operating system. In multimedia operating system the file system stores the files with following issues [17]: (1) physical storage device (2) contiguous storage of files that improves

the throughput at expense of management issues. (3)The disk scheduling to reduce the seek operation and fair disk [11].

For the multimedia disk scheduling the traditional disk scheduling approaches the substituted by EDF, SCAN-EDF, group-sweeping scheduling, mixed strategy, and continuous media file system. A life span of file system is longer than the execution of the program. The integration of discrete and continuous data needs additional resources. The time requirement is very important in multimedia applications. Thus disk scheduling techniques pay major role providing multimedia data [9].

6.1. Multimedia File System

Due to the need of immense storage and continuous media requirements the traditional tape drives are not feasible to store multimedia data [25]. But storage devices such as CD-ROM, RW-CDROM are used. The continuous media of multimedia system is differing from discrete data in the following conditions [11]:

Real time characteristics: The retrieval, computation and presentation time of continuous media are time independent.

File size: compared to text and graphics, video and audio have very large storage space requirements. Since the file system has to store information ranging from small unstructured units like text files to large, highly structured data units like video and associated audio, it has to organize the data on disk in a way that efficiently uses the limited storage.

Multiple data streams: a multimedia system has to support different media at the same time. It not only has to ensure that each medium gets a sufficient share of the resources, but it also has to consider the tight relationships between different streams arriving from different sources. The retrieval of a movie, for example, requires the processing and synchronization of audio and video.

7. Disk Scheduling Algorithms

The overall goal of disk scheduling in multimedia systems is to meet the deadlines of all time-critical tasks. The goal of keeping the necessary buffer space requirements low is loosely related. As many streams as possible should be served concurrently, but aperiodic requests should also be schedulable without delaying their service for an infinite amount of time. The scheduling algorithm has to find a balance between time constraints and efficiency [19].

7.1 Earliest Deadline First

This algorithm is used in CPU scheduling and also it is used in disk scheduling. In this algorithm at every new ready state, the scheduler selects from the tasks that are ready and not fully processed the one with the earliest deadline. The requested resource is assigned to the selected task. At the arrival of any new task, EDF must be computed immediately, heading to a new order, i.e. the running task is preempted and the new task is scheduled according to its deadline. The new task is processed immediately if its deadline is earlier than that of the interrupted task. The processing of the interrupted task is continued according to the EDF algorithm later on. EDF is not only an algorithm for

periodic tasks, but also for tasks with arbitrary requests, deadlines and service execution times[19].In file systems the block of the stream with the nearest deadline would be read first. This results in poor throughput and an excessive seek time; no buffer space is optimized. Further, as EDF is usually applied as a preemptive scheduling scheme, the costs for preemption of a task and scheduling of another task are considerable.

7.2 SCAN-Earliest Deadline First Algorithm

The SCAN-EDF is the combination of SCAN and EDF mechanism. The nearest seek time will be read first [24]. If there is more than read with same seek time, it will be read based on SCAN direction. This optimization can be applied to the reads with same seek time. When there is more than one reads with same deadline, they are grouped on the basis of their finish time. Buffer space is not optimized. The throughput is larger than EDF.

7.3 Group Sweeping Scheduling

With Group Sweeping scheduling, requests are served in cycle or in round robin manner. To reduce the disk arm movements, a set of n streams is divided into g groups. Individual streams are within a group are served according to scan technology. There is no fixed time to serve the streams. If the SCAN scheduling is applied to the streams without grouping the playout of a stream cannot be until the previous stream finish its payload. As the buffers can be reused for each group the playout of each stream starts at end where the first retrieval takes place.

7.4 Mixed strategy

The mixed strategy is based on the shortest seek and the balanced strategy [25].The data retrieved from the disk is transferred to the buffer allocated for the respective data stream. In this algorithm the data block which is closest is served first. The employment of shortest seek follows two criteria (1) the number of buffers for all the processes should be balanced and (2) overall require bandwidth should be sufficient to all active processes.

7.5 Continuous Media File System

The Continuous Disk Scheduling is a non preemptive scheduling scheme designed for the continuous media file system. The notion of slack time is introduced here. The slack time is the time duration for which the CMFS is free to do non real time operations.

Table 3: Comparison of the disk scheduling techniques

identify the major approaches and to present at least one representative for each. The various currently available CPU scheduling mechanisms, along with specialized CPU schedulers are discussed.

The memory management techniques along with VoD memory model give an overview of the memory management in Multimedia operating systems. The device management techniques are briefed. We have also discussed the various file management techniques available. The

Properties	Real time processing	Throughput	Seek time	Buffer
EDF	Yes	Poor	Excessive	No optimization
SCAN-EDF	Yes	Higher	Minimum	No optimization
GSW	Yes	Higher	Minimum	No optimization
Mixed Strategy	Yes	Maximum	Minimum	Optimization of buffer
CMFS	No	Maximum	Minimum	Optimization

various disk scheduling algorithms like EDF, SCAN-EDF, Group Sweeping, mixed strategy and continuous Media file system are also discussed along with their comparisons.

References

- [1] C. Lu, J. A. Stankovic, G. Tav, and S. H. Son. "The design and evaluation of a feedback control EDF scheduling algorithm," In proc of the 20th IEEE real time systems symp., Dec 1999.
- [2] D. C. Steere, A. Goel, J. Gruenburg, D. McNamee, C. Pu, and J. Walpole, "A feedback-driven proportion allocator for real rate scheduling," In proc of the 3rd symp on operating systems design and implementation , new Orleans, LA, Feb1999.
- [3] B. Ford and S. Susarla, "CPU inheritance scheduling," In proc of the 2nd symp on operating systems design and implementation, Oct 1996.
- [4] K. J. Duda and D. C. Chriton, "Borrowed- virtual time scheduling supporting latency – sensitive threads in a general purpose scheduler," In proc of the 17th ACM symp on operating systems principles, Kiawah Island, Dec 1999.
- [5] T. Plagemann, V. Goebel, P. Halvorsen, O. Anshus. "Operating system support for multimedia systems," Computer Communications, 23 (2006), 267–289.
- [6] T. Plagemann, V. Goebel, P. Halvorsen, O. Anshus. "Operating system support for multimedia systems," Computer Communications 23 (2000) 267–289.
- [7] Pawan Goyal, Xingang Guo, and Harrick M. Vin. "A Hierarchical CPU Scheduler for Multimedia Operating Systems," Proceedings of the second USENIX symposium on Operating systems design and implementation, pp: 107-121, 1996.
- [8] Daniel Alexander Taranovsky, "CPU Scheduling in Multimedia Operating Systems,"Research Report, 1999.
- [9] Gifford, D W and O'Toole, J W 'Intelligent file systems for object repositories', Operating Systems of the 90s and Beyond, Int. Workshop, Dagstuhl Castle, Germany (2007), 20-24.
- [10] P. A. Janson, "Operating Systems, Structures and Mechanisms," Academic Press, Orlando, FL

8. Conclusion

This article gives an overview of the OS support for multimedia applications. This is an active area, and a lot of valuable research results have been published. Thus, we have not discussed or cited *all* recent results, but tried to

- Tanenbaum, A S Operating System, Design and Implementation. Prentice-Hall, 2008.
- [11] Cliffs Englewood, S. J. Mullender, 'Systems of the nineties-Distributed multimedia systems; systems of the 90s and beyond', Int. Workshop, 1999.
- [12] Castle Dagstuhl, R. Steinmetz, "Data compression in multimedia computing: principles and techniques," *Multimedia Systems*, Vol 1 No 4, pp 166-172.
- [13] R. Steinmetz, "Data compression in multimedia computing: standards and systems," *Multimedia Systems*, Vol 1 No 5, 1994.
- [14] Lougher, P and Shepherd, D 'The design of a storage service for continuous media', *The Computer J*, Vol 36 No 1, pp 32-42, 1993.
- [15] J. Gemmell, and S. Christodoulakis, "Principles of delay sensitive multimedia data storage and retrieval," *ACM Trans. Infor. Syst.*, Vol IO No 1, January 1992.
- [16] Rangan, P V, Klppner, T and Vin, H W, 'Techniques for efficient storage of digital video and audio', *Proc. Workshop on Multimedia Information Systems*, Tempe, AZ, February 2002.
- [17] P. V. Rangan, and H. M. Vin, "Designing file systems for digital video and audio," *Proc. 13th ACM Symposium on Operating Systems Principles*, Monterey CA *Operating Systems Review*, Vol25 No 5, Oct., 1991.
- [18] P. V. Rangan, and H. M. Vin, "Techniques for efficient storage of digital video and audio," *Comput. Commun.*, Vol 16, pp 168-176, 2003.
- [19] A. Karmouch, Wang, and Yea, "Design and Analysis of a Storage Retrieval Model for Audio and Video Data," *Technical Report*, *Multimedia Information Systems*, Department of Electrical Engineering, University of Ottawa, Canada, 1994.
- [20] M. L. Dertouzos, "Control robotics," *The Procedural Control of Physical Processing*. Information Processing 74, North Holland, pp 807-813.
- [21] S. Krakowiak, "Principles of Operating Systems," MIT Press, Cambridge, MA, 2008.
- [22] J. Peterson, and A. Silberschatz, "Operating System Concepts," Addison-Wesley, Reading, MA, 1983.
- [23] R. Steinmetz, and K. Nahrstedt, "The Fundamentals in Multimedia Systems," Prentice-Hall, Englewood Cliffs, NJ, February 1995.
- [24] Y. N. Doganata, and A. Tantawy, "A cost/performance study of video servers with hierarchical storage," *IEEE Proc. Int. Conf. Multimedia Computing and Systems*, Boston, MA, 2005.
- [25] Ralf Steinmetz, "multimedia file systems: approaches for continuous media disk scheduling," *computer communications*, volume 18, number 3, 1995.
- [26] Sity Jason Neih and Monica S. Lam, "Computer Systems Laboratory," Stanford University, implementation and Evaluation of SMART:A Scheduler for Multimedia Applications.
- [27] T. D. C. Little, and A. Ghafoor. "Scheduling of Bandwidth-Constrained Multimedia Traffic," *Second International Workshop*, November 1991.

Survey on Distributed Operating Systems: A Real Time Approach

Shailesh Khapre, Rayer Jean, J. Amudhavel, D. Chandramohan, P. Sujatha and V. Narasimhulu

Department of Computer Science, Pondicherry Central University,
Pondicherry - 605014, India.

{shaileshkxaprekl, jeanrayar, amudhavel86, chandrumeister, spothula, narasimhavasi}@gmail.com

Abstract: Today's typical computing environment has changed from mainframe systems to small computing systems that often cooperate via communication networks. Distributed Operating Systems Concepts and Design addresses the organization and principles of distributed computing systems. Although it does not concentrate on any particular operating system or hardware, it introduces the major concepts of distributed operating systems without requiring that readers know all the theoretical or mathematical fundamentals. Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed.

Keywords: Distributed Systems, Modern Operating Systems.

1. Introduction

Everyone agrees that distributed systems are going to be very important in the future. Unfortunately, not everyone agrees on what they mean by the term "distributed system." In this paper we present a view point widely held within academia about what is and is not a distributed system, we discuss numerous interesting design issues concerning them, and finally we conclude with a fairly close look at some experimental distributed systems that are the subject of ongoing research at universities[1]. A distributed operating system is one that looks to its users like an ordinary centralized operating system but runs on multiple, independent central processing units (CPUs). The key concept here is transparency. In other words, the use of multiple processors should be invisible (transparent) to the user. Another way of expressing the same idea is to say that the user views the system as a "virtual uniprocessor," not as a collection of distinct machines.

To make the contrast with distributed operating systems stronger, let us briefly look at another kind of system, which we call a "network operating system." A typical configuration for a network operating system would be a collection of personal computers along with a common printer server and file server [35] for archival storage, all tied together by a local network. Users are typically aware of where each of their files are kept and must move files between machines with explicit "file transfer" commands, instead of having file placement managed by the operating system. The system has little or no fault tolerance [3][6][17]; if 1 percent of the personal computers crashes, 1 percent of

the users are out of business, instead of everyone simply being able to continue normal work, albeit with 1 percent worse performance.

1.1 Goals and Problem

A more fundamental problem in distributed systems is the lack of global state information. It is generally a bad idea to even try to collect complete information about any aspect of the system in one table. Lack of up-to-date information makes many things much harder. It is hard to schedule the processors optimally if you are not sure how many are up at the moment. Many people, however, think that these obstacles can be overcome in time, so there is great interest in doing research on the subject.

2. Network Operating System

Before starting our discussion of distributed operating systems, it is worth first taking a brief look at some of the ideas involved in network operating systems, since they can be regarded as primitive forerunners. Although attempts to connect computers together have been around for decades, networking really came into the limelight with the ARPANET in the early 1970s. The original design did not provide for much in the way of a network operating system. Instead, the emphasis was on using the network as a glorified telephone line to allow remote login and file transfer. Later, several attempts were made to create network operating systems, but they never were widely used. In more recent years, several research organizations have connected collections of minicomputers running the UNIX operating system into a network operating system, usually via a local network [9] [19] [29] gives a good survey of these systems, which we shall draw upon for the remainder of this section. As we said earlier, the key issue that distinguishes a network operating system from a distributed one is how aware the users are of the fact that multiple machines are being used. This visibility occurs in three primary areas: the file system, protection, and program execution. Of course, it is possible to have systems that are highly transparent in one area and not at all in the other, which leads to a hybrid form.

2.1 File System

When connecting two or more distinct systems together, the first issue that must be faced is how to merge the file systems [18] [19] [22] [37]. Three approaches have been tried.

The first approach is not to merge them at all. Going this route means that a program on machine A cannot access files on machine B by making system calls. Instead, the user must run a special file transfer program that copies the needed remote files to the local machine, where they can then be accessed normally. Sometimes remote printing and mail is also handled this way. One of the best-known examples of networks that primarily supports file transfer [11] and mail via special programs, and not system call access to remote files, is the UNIX "uucp" program, and its network, USENET.

The next step upward in the direction of a distributed file system is to have adjoining file systems. In this approach, programs on one machine can open files on another machine by providing a path name telling where the file is located. For example, one could say `open('/machine1/pathname', READ); open("machine/pathname", READ); open('/. /machine1/pathname', READ);` The latter naming scheme is used in the Newcastle Connection [19] and is derived from the creation of a virtual "super directory" above the root directories of all the connected machines. Thus `"/."` means start at the local root directory and go upward one level (to the super directory), and then down to the root directory of machine." In Figure 1, the root directory of three machines, A, B, and C are shown, with a super directory above them. To access file x from machine C, one could say `open('/. /C/x', READ-ONLY)`. In the Newcastle system, the naming tree is actually more general, since "machine 1" may really be any directory, so one can attach a machine as a leaf anywhere in the hierarchy, not just at the top.

The third approach is the way it is done in distributed operating systems, namely, to have a single global file system visible from all machines. When this method is used, there is one "bin" directory for binary programs, one password file, and so on. When a program wants to read the password file it does something like `open('/etc/passwd', READ-ONLY)` without reference to where the file is. It is up to the operating system to locate the file and arrange for transport of data as they are needed. LOCUS is an example of a system using this approach. The convenience of having a single global name space is obvious. In addition, this approach means that the operating system is free to move files around among machines to keep all the disks equally full and busy, and that the system can maintain.

Replicated copies of files if it so chooses. When the user or program must specify the machine name, the system cannot decide on its own to move a file to a new machine because that would change the (user visible) name used to access the file. Thus in network operating system, control over file placement must be done manually by the users, whereas in a distributed operating system it can be done automatically by the system itself.

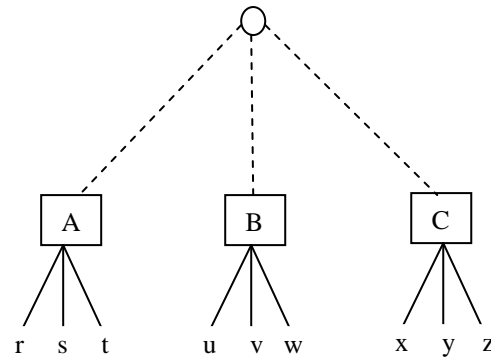


Figure 1. A (virtual) superdirectory above the root directory provides access to remote files.

2.2 Protection

Closely related to the transparency of the file system is the issue of protection. UNIX and many other operating systems assign a unique internal identifier to each user. Each file in the file system has a little table associated with it (called an i-node in UNIX) telling who the owner is, where the disk blocks are located, etc. If two previously independent machines are now connected, it may turn out that some internal User Identifier (UID), for example, number 12, has been assigned to a different user on each machine. Consequently, when user 12 tries to access a remote file, the remote file system cannot see whether the access is permitted since two different users have the same UID. One solution to this problem is to require all remote users wanting to access files on machine X to first log onto X using a user name that is local to X. When used this way, the network is just being used as a fancy switch to allow users at any terminal to log onto any computer, just as a telephone company switching center allows. A better approach is to have the operating system provide a mapping between UIDs, so that when a user with UID 12 on his or her home machine accesses a remote machine on which his or her UID is 15, the remote machine treats all accesses as though they were done by user 15. This approach implies that sufficient tables are provided to map each user from his or her home (machine, UID) pair to the appropriate UID for any other machine (and that messages cannot be tampered with).any subscriber to call any other subscriber. This solution is usually inconvenient for people and impractical for programs, so something better is needed. The next step up is to allow any user to access files on any machine without having to log in, but to have the remote user appear to have the UID corresponding to "GUEST" or "DEMO" or some other publicly known login name. Generally such names have little authority and can only access files that have been designated as readable or writable by all users. In a true distributed system there should be a unique UID for every user, and that UID should be valid on all machines without any mapping. In this way no protection problems arise on remote accesses to files; as far as protection goes, a remote access can be treated like a local access with the same UID. The protection issue makes the difference between a network operating system and a distributed one clear: In one case there are various machines, each with its own user-to-UID mapping and in the other there is a single, system wide mapping that is valid everywhere.

2.3 Execution Location

Program execution is the third area in which machine boundaries are visible in network operating systems. When a user or a running program wants to create a new process, where is the process created? At least four schemes have been used thus far. The first of these is that the user simply says "CREATE PROCESS" in one way or another, and specifies nothing about where. Depending on the implementation, this can be the best or the worst way to do it. In the most distributed case, the system chooses a CPU by looking at the load, location of files to be used, etc. In the least distributed case, the system always runs the process on one specific machine (usually the machine on which the user is logged in). The second approach to process location is to allow users to run jobs on any machine by first logging in there. In this model, processes on different machines cannot communicate or exchange data, but a simple manual load balancing is possible. The third approach is a special command that the user types at a terminal to cause a program to be executed on a specific machine. A typical command might be `remote vax4 who to run program on machine vax4`. In this arrangement, the environment of the new process is the remote machine. In other words, if that process tries to read or write files from its current working directory, it will discover that its working directory is on the remote machine, and that files that were in the parent process's directory are no longer present. Similarly, files written in the working directory will appear on the remote machine, not the local one. The fourth approach is to provide the "CREATE PROCESS" system call with a parameter specifying where to run the new process, possibly with a new system call for specifying the default site. As with the previous method, the environment will generally be the remote machine. In many cases, signals and other forms of interprocess communication between processes do not work properly among processes on different machines. A final point about the difference between network and distributed operating systems is how they are implemented. A common way to realize a network operating system is to put a layer of software on top of the native operating systems of the individual machines. For example, one could write a special library package that would intercept all the system calls and decide whether each one was local or remote [19] Although most system calls can be handled this way without modifying the kernel, invariably there are a few things, such as interprocess signals, interrupt characters (e.g., BREAK) from the keyboard, etc., that are hard to get right. In a true distributed operating system one would normally write the kernel from scratch.

1.4 An Example: The Sun Network File System

To provide a contrast with the true distributed systems described later in this paper, in this section we look briefly at a network operating system that runs on the Sun Microsystems' workstations. These work stations are intended for use as personal computers. Each one has a 68000 series CPU, local memory, and a large bit-mapped display. Workstations can be configured with or without local disk, as desired. All the workstations run a version of 4.2BSD UNIX specially modified for networking. This arrangement is a classic example of a network operating system: Each computer runs a traditional operating system, UNIX, and each has its own user(s), but with extra features added to make networking more convenient. During its

evolution the Sun system has gone through three distinct versions, which we now describe. In the first version each of the work-stations was completely independent from all the others, except that a program rep was provided to copy files from one work-station to another. By typing a command such as `rep M1:/usr/jim/file.c M2:/usr/ast/f.c` it was possible to transfer whole files from one machine to another. In the second version, Network Disk (ND), a network disk server was provided to support diskless workstations. Disk space on the disk server's machine was divided into disjoint partitions, with each partition acting as the virtual disk for some (diskless) workstation. Whenever a diskless workstation needed to read a file, the request was processed locally until it got down to the level of the device driver, at which point the block needed was retrieved by sending a message to the remote disk server. In effect, the network was merely being used to simulate a disk controller. With this network disk system, sharing of disk partitions was not possible. The third version, the Network File System (NFS), allows remote directories to be mounted in the local file tree on any workstation. By mounting, say, a remote directory "dot" on the empty local directory "/usr/doc," all subsequent references to "/usr/doc" are automatically routed to the remote system. Sharing is allowed in NFS, so several users can read files on a remote machine at the same time. To prevent users from reading other people's private files, a directory can only be mounted remotely if it is explicitly exported by the workstation it is located on. A directory is exported by entering a line for it in a file "/etc/exports." To improve performance of remote access, both the client machine and server machine do block caching. Remote services can be located using a Yellow Pages server that maps service names onto their network locations. The NFS is implemented by splitting the operating system up into three layers. The top layer handles directories, and maps each path name onto a generalized i-node called a vnode consisting of a (machine, i-node) pair, making each vnode globally unique.

Vnode numbers are presented to the middle layer, the virtual file system (VFS). This layer checks to see if a requested vnode is local or not. If it is local, it calls the local disk driver or, in the case of an ND partition, sends a message to the remote disk server. If it is remote, the VFS calls the bottom layer with a request to process it remotely. The bottom layer accepts requests for accesses to remote vnodes and sends them over the network to the bottom layer on the serving machine. From there they propagate upward through the VFS layer to the top layer, where they are reinjected into the VFS layer. The VFS layer sees a request for a local vnode and processes it normally, without realizing that the top layer is actually working on behalf of a remote kernel. The reply retraces the same path in the other direction. The protocol between workstations has been carefully designed to be robust in the face of network and server crashes. Each request completely identifies the file (by its vnode), the position in the file, and the byte count. Between requests, the server does not maintain any state information about which files are open or where the current file position is. Thus, if a server crashes and is rebooted, there is no state information that will be lost. The ND and NFS facilities are quite different and can both be used on the same workstation without conflict. ND works at a low level

and just handles remote block I/O without regard to the structure of the information on the disk. NFS works at a much higher level and effectively takes requests appearing at the top of the operating system on the client machine and gets them over to the top of the operating system on the server machine, where they are processed in the same way as local requests.

3. Design Issues

Now we turn from traditional computer systems with some networking facilities added on to systems designed with the intention of being distributed. In this section we look at five issues that distributed systems' designers are faced with: communication primitives, naming and protection, resource management, fault tolerance [4][5][6], services to provide. Although no list could possibly be exhaustive at this early stage of development, these topics should provide a reasonable impression of the areas in which current research is proceeding.

3.1 Communication Primitives

The computers forming a distributed system normally do not share primary memory, and so communication via shared memory techniques such as semaphores and monitors is generally not applicable. Instead, message passing in one form or another is used [23]. One widely discussed framework for message-passing systems is the ISO OSI reference model, which has seven layers, each performing a well-defined function. The seven layers are the physical layer, data-link layer, network layer, transport layer, session layer, presentation layer, and application layer. By using this model it is possible to connect computers with widely different operating systems, character codes, and ways of viewing the world. Unfortunately, the overhead created by all these layers is substantial. In a distributed system consisting primarily of huge mainframes from different manufacturers, connected by slow leased lines (say, 56 kilobytes per second), the overhead might be tolerable. Plenty of computing capacity would be available for running complex protocols, and the narrow bandwidth means that close coupling between the systems would be impossible anyway. On the other hand, in a distributed system consisting of identical microcomputers connected by a lo-megabyte-per second or faster local network, the price of the ISO model is generally too high. Nearly all the experimental distributed systems discussed in the literature thus far have opted for a different, much simpler model, so we do not mention the ISO model further in this paper.

3.2 Message Passing

The model that is favored by researchers in this area is the client-server model, in which a client process wanting some service (e.g., reading some data from a file) sends a message to the server and then waits for a reply message, as shown in Figure 2. In the most naked form the system just provides two primitives: SEND and RECEIVE. The SEND primitive specifies the destination and provides a message; the RECEIVE primitive tells from whom a message is desired (including "anyone") and provides a buffer where the

incoming message is to be stored. No initial setup is required, and no connection is established, hence no tear down is required.

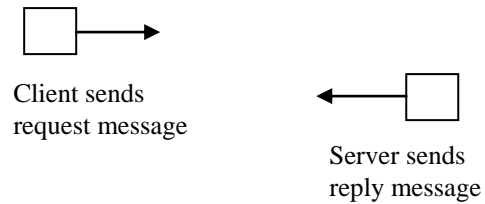


Figure 2. Client-server model of communication

Precisely what semantics these primitives ought to have has been a subject of much controversy among researchers. Two of the fundamental decisions that must be made are unreliable versus reliable and no blocking versus blocking primitives. At one extreme, SEND can put a message out onto the network and wish it good luck. No guarantee of delivery is provided, and no automatic retransmission is attempted by the system if the message is lost. At the other extreme, SEND can handle lost messages, retransmissions, and acknowledgments internally, so that when SEND terminates, the program is sure that the message has been received and acknowledged.

Blocking versus Non blocking Primitives: The other choice is between no blocking and blocking primitives. With nonblocking primitives, SEND returns control to the user program as soon as the message has been queued for subsequent transmission (Or a copy made). If no copy is made, any Changes the program makes to the data before or (heaven forbid) while they are being sent are made at the program's peril. When the message has been transmitted (or copied to a safe place for subsequent transmission), the program is interrupted to inform it that the buffer may be reused. The corresponding RECEIVE primitive signals a willingness to receive a message and provides a buffer for it to be put into. When a message has arrived, the program is informed by interrupt, or it can poll for status continuously or go to sleep until the interrupt arrives. The advantage of these non blocking primitives is that they provide the maximum flexibility: Programs can compute and perform message I/O in parallel in any way they want. Non blocking primitives also have a disadvantage: They make programming tricky and difficult. Irreproducible, timing-dependent programs are painful to write and awful to debug. Consequently, many people advocate sacrificing some flexibility and efficiency by using blocking primitives. A blocking SEND does not return control to the user until the message has been sent (unreliable blocking primitive) or until the message has been sent and an acknowledgment received (reliable blocking primitive). Either way, the program may immediately modify the buffer without danger. A blocking RECEIVE does not return control until a message has been placed in the buffer. Reliable and unreliable RECEIVES differ in that the former automatically acknowledges receipt of a message, whereas the latter does not. It is not reasonable to combine a reliable SEND with an unreliable RECEIVE, or vice versa; so the system designers must make a choice and provide one set or the other. Blocking and non-blocking primitives do not conflict, so

there is no harm done if the sender uses one and the receiver the other. Receiver, although buffered message passing can be implemented in many ways, a typical approach is to provide users with a system call `CREATEBUF`, which creates a kernel buffer, sometimes called a mailbox, of a user-specified size. To communicate, a sender can now send messages to the receiver's mailbox, where they will be buffered until requested by the receiver. Buffering is not only more complex (creating, destroying, and generally managing the mailboxes), but also raises issues of protection, the need for special high-priority interrupt messages, what to do with mail-boxes owned by processes that have been killed or died of natural causes, and more. A more structured form of communication is achieved by distinguishing requests from replies. With this approach, one typically has three primitives: `SEND-GET`, `GET-REQUEST`, and `SEND-REPLY`. `SEND-GET` is used by clients to send requests and get replies. It combines a `SEND` to a server with a `RECEIVE` to get the server's reply. `GET-REQUEST` is done by servers to acquire messages containing work for them to do. When a server has carried the work out, it sends a reply with `SEND-REPLY`. By thus restricting the message traffic and using reliable, blocking primitives, one can create some order in the chaos.

3.3 Remote Procedure Call (RPC)

The next step forward in message-passing systems is the realization that the model of "client sends request and blocks until server sends reply" looks very similar to a traditional procedure call from the client to the server. This model has become known in the literature as "remote procedure call" and has been widely discussed [10] [12]. The idea is to make the semantics of inter-machine communication as similar as possible to normal procedure calls because the latter is familiar and well understood, and has proved its worth over the years as a tool for dealing with abstraction. It can be viewed as a refinement of the reliable, blocking `SEND-GET`, `GET-REQUEST`, `SENDREP` primitives, with a more user-friendly syntax. The remote procedure call can be organized as follows. The client (calling program) makes a normal procedure call, say, `p(x, y)` on its machine, with the intention of invoking the remote procedure `p` on some other machine. A dummy or stub procedure `p` must be included in the caller's address space, or at least be dynamically linked to it upon call. This procedure, which may be automatically generated by the compiler, collects the parameters and packs them into a message in a standard format. It then sends the message to the remote machine (using `SEND-GET`) and blocks, waiting for an answer (see Figure 3). At the remote machine, another stub procedure should be waiting for a message using `GET-REQUEST`. When a message comes in, the parameters are unpacked by an input-handling procedure, which then makes the local call `p(x, y)`. The remote procedure `p` is thus called locally, and so its normal assumptions about where to find parameters, the state of the stack, etc., are identical to the case of a purely local call. The only procedures that know that the call is remote are the stubs, which build and send the message on the client side and disassemble and make the call on the server side. The result of the procedure call follows an analogous path in the reverse direction.

Although at first glance the remote procedure call model seems clean and simple, under the surface there are several problems. One problem concerns parameter (and result) passing. In most programming languages, parameters can be passed by value or by reference. Passing value parameters over the network is easy; the stub just copies them into the message and off they go. Passing reference parameters (pointers) over the network is not so easy. One needs a unique, system wide pointer for each object so that it can be remotely accessed. For large objects, such as files, some kind of capability mechanism [36] could be set up, using capabilities as pointers.

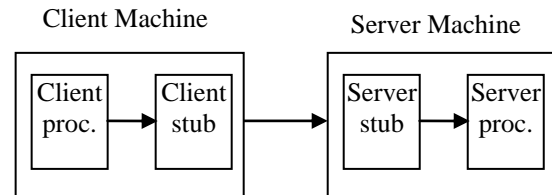


Figure 3. Remote procedure call.

Leans, the amount of overhead and mechanism needed to create a capability and send it in a protected way is so large that this solution is highly undesirable. Still another problem that must be dealt with is how to represent parameters and results in messages. This representation is greatly complicated when different types of machines are involved in a communication. A floating-point number produced on one machine is unlikely to have the same value on a different machine, and even a negative integer will create problems between the 1's complement and 2's complement machines. Converting to and from a standard format on every message sent and received is an obvious possibility, but it is expensive and wasteful, especially when the sender and receiver do, in fact, use the same internal format. If the sender uses its internal format (along with an indication of which format it is) and lets the receiver do the conversion, every machine must be prepared to convert from every other format. When a new machine type is introduced, much existing software must be upgraded. Any way it is done, with remote procedure call (RPC) or with plain messages, it is an unpleasant business. Some of the unpleasantness can be hidden from the user if the remote procedure call mechanism is embedded in a programming language with strong typing, so that the receiver at least knows how many parameters to expect and what types they have. In this respect, a weakly typed language such as C, in which procedures with a variable number of parameters are common, is more complicated to deal with. Still another problem with RPC is the issue of client-server binding. Consider, for example, a system with multiple file servers. If a client creates a file on one of the file servers, it is usually desirable that subsequent writes to that file go to the file server where the file was created. With mailboxes, arranging for this is straightforward. The client simply addresses the `WRITE` messages to the same mailbox that the `CREATE` message was sent to. Since each file server has its own mailbox, there is no ambiguity. When RPC is used, the situation is more complicated, since the entire client does is put a procedure call such as `write (File Descriptor, Buffer Address, Byte Count)`; in his program. RPC intentionally

hides all the details of locating servers from the client, but sometimes, as in this example, the details are important. In some applications, broadcasting and multicasting (sending to a set of destinations, rather than just one) is useful. For example, when trying to locate a certain person, process, or service, sometimes the only approach is to broadcast an inquiry message and wait for the replies to come back. RPC does not lend itself well to sending messages to sets of processes and getting answers back from some or all of them. The semantics are completely different. Despite all these disadvantages, RPC remains an interesting form of communication and much current research is being addressed toward improving it and solving the various problems discussed above.

3.4 Naming and Protection

All operating systems support objects such as files, directories, segments, mailboxes, processes, services, servers, nodes, and I/O devices. When a process wants to access one of these objects, it must present some kind of name to the operating system to specify which object it wants to access. In some instances these names are ASCII strings designed for human use; in others they are binary numbers used only internally. In all cases they have to be managed and protected from misuse.

3.4.1 Naming and Protection

Naming [33] can best be seen as a problem of mapping between two domains. For example, the directory system in UNIX provides a mapping between ASCII path names and i-node numbers. When an OPEN system call is made, the kernel converts the name of the file to be opened into its i-node number. Internal to the kernel, files are nearly always referred to by i-node number, not ASCII string. Just about all operating systems have something similar. In a distributed system a separate name server is sometimes used to map user-chosen names (ASCII strings) onto objects in an analogous way. Another example of naming is the mapping of virtual addresses onto physical addresses in a virtual memory system. The paging hardware takes a virtual address as input and yields a physical address as output for use by the real memory. In some cases naming implies only a single level of mapping, but in other cases it can imply multiple levels. For example, to use some service, a process might first have to map the service name onto the name of a server process that is prepared to offer the service. As a second step, the server would then be mapped onto the number of the CPU on which that process is running. The mapping need not always be unique, for example, if there are multiple processes prepared to offer the same service.

3.4.2 Name Servers

In centralized systems, the problem of naming can be effectively handled in a straightforward way. The system maintains a table or database providing the necessary name-to-object mappings. The most straightforward generalization of this approach to distributed systems is the single name server model. In this model, a server accepts names in one domain and maps them onto names in another domain. For example, to locate services in some distributed systems, one

sends the service name in ASCII to the name server, and it replies with the node number where that service can be found, or with the process name of the server process, or perhaps with the name of a mailbox to which requests for service can be sent. The name server's database is built up by registering services, processes, etc., that want to be publicly known. File directories can be regarded as a special case of name service. Although this model is often acceptable in a small distributed system located at a single site, in a large system it is undesirable to have a single centralized component (the name server) whose demise can bring the whole system to a grinding halt. In addition, if it becomes overloaded, performance will degrade. Furthermore, in a geographically distributed system that may have nodes in different cities or even countries, having a single name server will be inefficient owing to the long delays in accessing it. The next approach is to partition the system into domains, each with its own name server. If the system is composed of multiple local networks connected by gateways and bridges, it seems natural to have one name server per local network. One way to organize such a system is to have a global naming tree, with files and other objects having names of the form: /country/city/network/pathname. When such a name is presented to any name server, it can immediately route the request to some name server in the designated country, which then sends it to a name server in the designated city, and so on until it reaches the name server in the network where the object is located, where the mapping can be done. Telephone numbers use such a hierarchy, composed of country code, area code, exchange code (first three digits of telephone number in North America), and subscriber line number. Having multiple name servers does not necessarily require having a single, global naming hierarchy. Another way to organize the name servers is to have each one effectively maintain a table of, for example, (ASCII string, pointer) pairs, where the pointer is really a kind of capability for any object or domain in the system. When a name, say a/b/c, is looked up by the local name server, it may well yield a pointer to another domain (name server), to which the rest of the name, b/c, is sent for further processing (see Figure 4). This facility can be used to provide links (in the UNIX sense) to files or objects whose precise whereabouts is managed by a remote name server. Thus if a file foobar is located in another local network, n, with name server n.s, one can make an entry in the local name server's table for the pair (x, n.s) and then access x/foobar as though it were a local object. Any appropriately authorized user or process knowing the name x/foobar could make its own synonym s and then perform accesses using s/x/foobar. Each name server parsing a name that involves multiple name servers just strips off the first component and passes the rest of the name to the name server found by looking up the first component locally. A more extreme way of distributing the name server is to have each machine manage its own names. To look up a name, one broadcasts it on the network. At each machine, the incoming request is passed to the local name server, which replies only if it finds a match. Although broadcasting is easiest over a local network such as a ring net or CSMA net (e.g., Ethernet), it is also possible over store-and-forward packet switching networks such as the ARPANET [34]. Although the normal use of a name server is to map an ASCII string onto a binary

number used internally to the system, such as a process identifier or machine number, once in a while the inverse mapping is also useful. For example, if a machine crashes, upon rebooting it could present its (hard-wired) node number to the name server to ask what it was doing before the crash, that is, ask for the ASCII string corresponding to the service that it is supposed to be offering so that it can figure out what program to reboot.

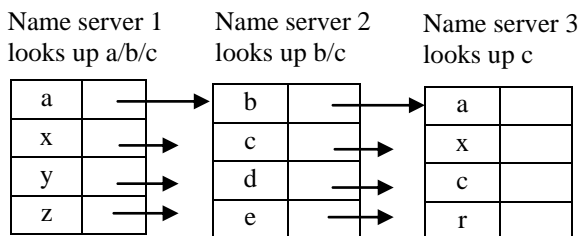


Figure 4. Distributing the lookup of a/b/c over three name servers

3.4.3 Process Allocation

One of the key resources to be managed in a distributed system is the set of available processors. One approach that has been proposed for keeping tabs on a collection of processors is to organize them in a logical hierarchy independent of the physical structure of the network, as in MICROS. This approach organizes the machines like people in corporate, military, academic, and other real-world hierarchies. Some of the machines are workers and others are managers. For each group of k workers, one manager machine (the “department head”) is assigned the task of keeping track of who is busy and who is idle. If the system is large, there will be an unwieldy number of department heads; so some machines will function as “deans,” riding herd on k department heads. If there are many deans, they too can be organized hierarchically, with a “big cheese” keeping tabs on k deans. This hierarchy can be extended ad infinitum, with the number of levels needed growing logarithmically with the number of workers. Since each processor need only maintain communication with one superior and k subordinates, the information stream is manageable [15]. An obvious question is, “What happens when a department head, or worse yet, a big cheese, stops functioning (crashes)?” One answer is to promote one of the direct subordinates of the faulty manager to fill in for the boss. The choice of which one can either be made by the subordinates themselves, by the deceased’s peers, or in a more autocratic system, by the sick manager’s boss. To avoid having a single (vulnerable) manager at the top of the tree, one can truncate the tree at the top and have a committee as the ultimate authority. When a member of the ruling committee malfunctions, the remaining members promote someone one level down as a replacement. Although this scheme is not completely distributed, it is feasible and works well in practice. In particular, the system is self-repairing, and can survive occasional crashes of both workers and managers without any long-term effects. In MICROS, the processors are mono-programmed, so if a job requiring S processes suddenly appears, the system must

allocate S processors for it. Jobs can be created at any level of the hierarchy. The strategy used is for each manager to keep track of approximately how many workers below it are available (possibly several levels below it). If it thinks that a sufficient number are available, it reserves some number R of them, where $R \geq S$, because the estimate of available workers may not be exact and some machines may be down. If the manager receiving the request thinks that it has too few processors available, it passes the request upward in the tree to its boss. If the boss cannot handle it either, the request continues propagating upward until it reaches a level that has enough available workers at its disposal. At that point, the manager splits the request into parts and parcels them out among the managers below it, which then do the same thing until the wave of scheduling requests hits bottom. At the bottom level, the processors are marked as “busy,” and the actual number of processors allocated is re-ported back up the tree. To make this strategy work well, R must be large enough so that the probability is high that enough workers will be found to handle the whole job. Otherwise, the request will have to move up one level in the tree and start all over, wasting considerable time and computing power. On the other hand, if R is too large, too many processors will be allocated, wasting computing capacity until word gets back to the top and they can be released. The whole situation is greatly complicated by the fact that requests for processors can be generated randomly anywhere in the system, so at any instant, multiple requests are likely to be in various stages of the allocation algorithm, potentially giving rise to out-of-date estimates of available workers, race conditions, deadlocks, and more. In Van, a mathematical analysis of the problem is given and various other aspects not described here are covered in detail.

3.4.4 Scheduling

The hierarchical model provides a general model for resource control but does not provide any specific guidance on how to do scheduling. If each process uses an entire processor (i.e., no multiprogramming), and each process is independent of all the others, any process can be assigned to any processor at random. However, if it is common that several processes are working together and must communicate frequently with each other, as in UNIX pipelines or in cascaded (nested) remote procedure calls, then it is desirable to make sure that the whole group runs at once. In this section we address that issue. Let us assume that each processor can handle up to N processes.

If there are plenty of machines and N is reasonably large, the problem is not finding a free machine (i.e., a free slot in some process table), but something more subtle. The basic difficulty can be illustrated by an example in which processes A and B run on one machine and processes C and D run on another. Each machine is time shared in, say, 100-millisecond time slices, with A and C running in the even slices, and B and D running in the odd ones, as shown in Figure 5a. Suppose that A sends many messages or makes many remote procedure calls to D. During time slice 0, A starts up and immediately calls D, which unfortunately is not running because it is now C’s turn. After 100 milliseconds, process switching takes place, and D gets A’s message, carries out the work, and quickly replies. Because B is now running, it will be another 100 milliseconds before A gets

the reply and can proceed. The net result is one message exchange every 200 milliseconds. What is needed is a way to ensure that processes that communicate frequently run simultaneously. Although it is difficult to determine dynamically the inter process communication patterns, in many cases a group of related processes will be started off together.

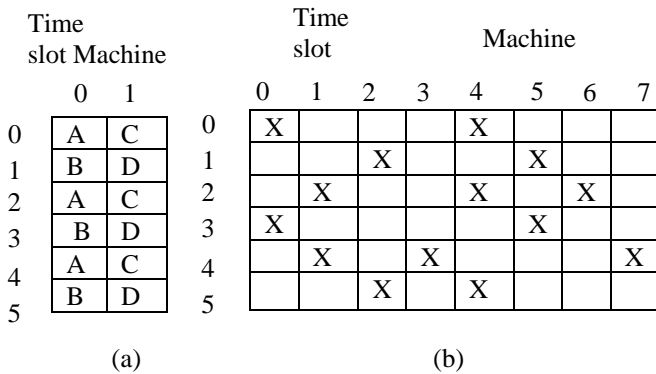


Figure 5. (a) Two jobs running out of phase with each other. (b) Scheduling matrix for eight machines, each with six time slots. The X's indicated allocated slots.

For example, it is usually a good bet that the filters in a UNIX pipeline will communicate with each other more than they will with other, previously started processes. Let us assume that processes are created in groups, and that intergroup communication is much more prevalent than intergroup communication. Let us further assume that a sufficiently large number of machines are available to handle the largest group, and that each machine is multiprogrammed with N process slots (N-way multiprogramming). Previous work has proposed several algorithms based on the concept of co-scheduling, which takes interprocess communication patterns into account while scheduling to ensure that all members of a group run at the same time. The first algorithm uses a conceptual matrix in which each column is the process table for one machine, as shown in Figure 5b. Thus, column 4 consists of all the processes that run on machine 4. Row 3 is the collection of all processes that are in slot 3 of some machine, starting with the process in slot 3 of machine 0, then the process in slot 3 of machine 1, and so on. The gist of his idea is to have each processor use a round-robin scheduling algorithm with all processors first running the process in slot 0 for a fixed period, then all processors running the process in slot 1 for a fixed period, etc. A broadcast message could be used to tell each processor when to do process switching, to keep the time slices synchronized. By putting all the members of a process group in the same slot number, but on different machines, one has the advantage of N-fold parallelism, with a guarantee that all the processes will be run at the same time, to maximize communication throughput. Thus in Figure 5b, four processes that must communicate should be put into slot 3, on machines 1, 2, 3, and 4 for optimum performance. This scheduling technique can be combined with the hierarchical model of process management used in MICROS by having each department head maintain the matrix for its workers, assigning processes to slots in the matrix and broadcasting

time signals. Ouster out also described several variations to this basic method to improve performance. One of these breaks the matrix into rows and concatenates the rows to form one long row. With k machines, any k consecutive slots belong to different machines. To allocate a new process group to slots, one lays a window k slots wide over the long row such that the leftmost slot is empty but the slot just outside the left edge of the window is full. If sufficient empty slots are present in the window, the processes are assigned to the empty slots; otherwise the window is slid to the right and the algorithm repeated. Scheduling is done by starting the window at the left edge and moving rightward by about one window's worth per time slice, taking care not to split groups over windows. Usterhout's paper discusses these and other methods in more detail and give some performance results.

3.4.5 Load Balancing

The goal of Usterhout's work is to place processes that work together on different processors, so that they can all run in parallel. Other researchers have tried to do precisely the opposite, namely, to find sub-sets of all the processes in the system that are working together, so that closely related groups of processes can be placed on the same machine to reduce inter process communication costs [30] [31] [32]. Yet other researchers have been concerned primarily with load balancing, to prevent a situation in which some processors are overloaded while others are empty [8] [38]. Of course, the goals of maximizing throughput, minimizing response time, and keeping the load uniform are to some extent in conflict, so many of the researchers try to evaluate different compromises and trade-offs. Each of these different approaches to scheduling makes different assumptions about what is known and what is most important. The people trying to cluster processes to minimize communication costs, for example, assume that any process can run on any machine, that the computing needs of each process are known in advance, and that the interprocess communication traffic between each pair of processes is also known in advance. The people doing load balancing typically make the realistic assumption that nothing about the future behavior of a process is known.

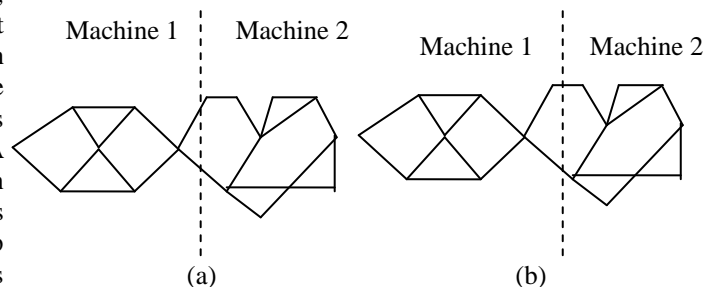


Figure 7. Two ways of statistically allocating processes (nodes in the graph) to machines. Arcs show which pairs of process communicate.

People making real systems, who care less about optimality than about devising algorithms that can actually be used, Let us now briefly look at each of these approaches.

Graph-Theoretic Models: If the system consists of a fixed number of processes, each with known CPU and memory requirements, and a known matrix giving the average amount of traffic between each pair of processes, scheduling can be attacked as a graph-theoretic problem. The system can be represented as a graph, with each process a node and each pair of communicating processes connected by an arc labeled i^{th} the data rate between them. The problem of allocating all the processes to k processors then reduces to the problem of partitioning the graph into k disjoint sub graphs, such that each sub graph meets certain constraints (e.g., total CPU and memory requirements below some limit). Arcs that are entirely within one sub graph represent internal communication within a single processor (= fast), whereas arcs that cut across sub graph boundaries represent communication between two processors (= slow). The idea is to find a partitioning of the graph that meets the constraints and minimizes the network traffic, or some variation of this idea. Figure 7a depicts a graph of interacting processors with one possible partitioning of the processes between two machines. Figure 7b shows a better partitioning, with less intermachine traffic, assuming that all the arcs are equally weighted. Many papers have been written on this subject [30] [31]. The results are somewhat academic, since in real systems virtually none of the assumptions (fixed number of processes with static requirements, known traffic matrix, error-free processors and communication) are ever met.

Heuristic Load Balancing: Here the idea is for each processor to estimate its own load continually, for processors to exchange load information, and for process creation and migration to utilize this information. Various methods of load estimation are possible. One way is just to measure the number of runnable processes on each CPU periodically and take the average of the last n measurements as the load. Another way [20] is to estimate the residual running times of all the processes and define the load on a processor as the number of CPU seconds that all its processes will need to finish. The residual time can be estimated mostly simply by assuming it is equal to the CPU time already consumed. Bryant and Finkel also discuss other estimation techniques in which both the number of processes and length of remaining time are important. When round-robin scheduling is used, it is better to be competing against one process that needs 100 seconds than against 100 processes that each need 1 second. Once each processor has computed its load, a way is needed for each processor to find out how everyone else is doing. One way is for each processor to just broadcast its load periodically. After receiving a broadcast from a lightly loaded machine, a processor should shed some of its load by giving it to the lightly loaded processor. This algorithm has several problems. First, it requires a broadcast facility, which may not be available. Second, it consumes considerable bandwidth for all the "here is my load" messages. Third, there is a great danger that many processors will try to shed load to the same (previously) lightly loaded processor at once. A different strategy [8] is for each processor periodically to pick another processor (possibly a neighbor, possibly at random) and exchange load information with it. After the exchange, the more heavily loaded processor can send processes to the other one until they are equally loaded. In this model, if 100 processes are suddenly created in an otherwise empty system, after one exchange we will have

two machines with 50 processes and after two exchanges most probably four machines with 25 processes. Processes diffuse around the network like a cloud of gas. Actually migrating running processes is trivial in theory, but close to impossible in practice. The hard part is not moving the code, data, and registers, but moving the environment, such as the current position within all the open files, the current values of any running timers, pointers or file descriptors for communicating with tape drives or other I/O devices, etc. All of these problems relate to moving variables and data structures related to the process that are scattered about inside the operating system. What is feasible in practice is to use the load information to create new processes on lightly loaded machines, instead of trying to move running processes. If one has adopted the idea of creating new processes only on lightly loaded machines, another approach, called bidding, is possible [40]. When a process wants some work done, it broadcasts a request for bids, telling what it needs (e.g., a 68000 CPU, 512K memory, floating point, and a tape drive). Other processors can then bid for the work, telling what their workload is, how much memory they have available, etc. The process making the request then chooses the most suitable machine and creates the process there. If multiple request-for-bid messages are outstanding at the same time, a processor accepting a bid may discover that the workload on the bidding machine is not what it expected because that processor has bid for and won other work in the meantime.

3.4.6 Distributed Deadlock Detection

Some theoretical work has been done in the area of detection of deadlocks in distributed systems. How applicable this work may be in practice remains to be seen. Two kinds of potential deadlocks are resource deadlocks and communication deadlocks. Resource deadlocks are traditional deadlocks, in which all of some set of processes are blocked waiting for resources held by other blocked processes. For example, if A holds X and B holds Y, and A wants Y and B wants X, a deadlock will result. In principle, this problem is the same in centralized and distributed systems, but it is harder to detect in the latter because there are no centralized tables giving the status of all resources. The problem has mostly been studied in the context of database systems [39]. The other kind of deadlock that can occur in a distributed system is a communication deadlock. Suppose A is waiting for a message from B and B is waiting for C and C is waiting for A. Then we have a deadlock. [21] present an algorithm for detecting (but not preventing) communication deadlocks. Very crudely summarized, they assume that each process that is blocked waiting for a message knows which process or processes might send the message. When a process logically blocks, they assume that it does not really block but instead sends a query message to each of the processes that might send it a real (data) message. If one of these processes is blocked, it sends query messages to the processes it is waiting for. If certain messages eventually come back to the original process, it can conclude that a deadlock exists. In effect, the algorithm is looking for a knot in a directed graph.

4. DOS Survey

4.1 Comparison Criteria

The main goal of distributed file systems (DFS) or distributed operating systems (DOS) is to provide some level of transparency to the users of a computer network.

We have tried to develop a scheme -- referred to as a catalog of criteria -- that allows us to describe the systems in an implementation independent way. The main questions to be answered are: What kind of transparency levels are provided, how each kind of transparency achieved is what kind of communication strategy has been proposed and finally, does the distributed character of the system allow increased availability and reliability. The last question leads us to an analysis of replication schemes used and to an evaluation of proposed failure handling/recovery strategies.

4.1.1 Transparency Levels

We distinguish five levels of transparency. We speak of location transparency existing, when a process requesting a particular network resource does not necessarily know where the resource is located. Access transparency gives a user access to both local and remote located resources in the same way.

For reasons of availability, resources are sometimes replicated. If a user does not know whether a resource has been replicated or not, replication transparency exists.

The problem of synchronization is well-known. In a distributed environment this problem arises in an extended form. Encapsulating concurrency control inside a proposed system is what is meant by concurrency transparency. This includes schemes that provide system-wide consistency as well as weaker schemes in which user interaction can be necessary to recreate consistency. The distinction between transaction strategies and pure semaphore-based techniques is introduced in a special evaluation.

The last level of transparency is failure transparency. Network link failures or node crashes are always present within a network of independent nodes. Systems that provide stable storage, failure recovery and some state information are said to be failure transparent.

4.1.2 Heterogeneity

This survey furthermore gives information on the heterogeneity of the systems, i. e., assumptions made about the hardware and which operating system is used and whether that O.S. is a modified or look-alike version of another O.S.

We describe the underlying network. Is it a LAN, if so, what kind of LAN, or have gateways [7] been developed to integrate the systems into a WAN world.

4.1.3 Changes Made

There are two main forms of implementing distribution. First, a new layer can be inserted on top of an existing operating system that handles requests and provides remote access as well as some of the transparency levels.

Second, the distributed system can be implemented as a new kernel that runs on every node. This differentiation is a first hint of how portable or compatible a system is [28]. Some systems do not distribute all kernel facilities to all nodes. Dedicated servers can be introduced (strict client/server model). Some systems distribute a small kernel

to all nodes and the rest of the utilities to special nodes (non-strict client/server model). Another group of systems are the so called integrated systems. In an integrated system each node can be a client, a server or both. This survey tries to describe these differences.

4.1.4 Communication Protocols

Message passing is the main form of communication (excepting multiprocessor systems which can use shared memory). We show which kind of protocols are used and describe specialized protocols if implemented.

4.1.5 Connection and RPC Facility

The kind of connection established by the (peer) communication partners is another important criteria. We distinguish between point-to-point connections (virtual circuits), datagram-style connections, and connections based on pipes or streams. If a remote procedure call (RPC) facility is provided we add this information as well.

4.1.6 Semantics

The users of a distributed system are interested in the way their services are provided and what their semantics are. We distinguish may-be (which means that the system guarantees nothing), at-least-once semantics (retrying to fulfill a service until acknowledged, sometimes done twice or more frequently), at-most-once semantics (mostly achieved by duplicate detection) and exactly-once semantics. The last kind is achieved by making a service an atomic issue (so called all-or-nothing principle).

4.1.7 Naming Strategy

We describe the naming philosophy and distinguish between object-oriented and traditional hierarchical naming conventions. Our overview includes the proposed name space itself as well as the mechanisms used to provide a system-spanning name space (e. g. mounting facilities or super root-approaches).

4.1.8 Security Issue

Security plays an important role within distributed systems, since the administration could possibly be decentralized and participating hosts cannot necessarily be trusted. Intruders may find it easy to penetrate a distributed environment. Therefore, sophisticated algorithms for encryption and authentication are necessary. We add four entries concerning this issue. First, encryption is used if no plain text will be exchanged over the communication media. Second, some systems make use of special hardware components to achieve security during the message transfer. Third, capabilities are provided that enable particular users access to resources in a secure and predefined way. Finally, we introduce the entry mutual authentication. This feature is provided if a sort of hand-shake mechanism is implemented that allows bilateral recognition of trustworthiness.

4.1.9 Failure handling

Failure handling/recovery is a very critical issue. Since some systems are designed to perform well in an academic environment and some systems are made highly reliable for

commercial use, trade-off decisions must be taken into account. We add the following four entries to our catalog of

balancing schemes to increase the system's performance. We include this issue in our survey.

Table 1: Table of comparison – Types of System & Transparency of Different Types of DOS's.

Name	Type of System		Transparency				
	DOS	DFS	locati- on	access	repli- cation	con- currency	failure
Accent	*		*	*			*
Alpine		*		*			
Amoeba	*		*	*		*	*
Andrew		*	*	*	*	*	
Argus	*		*	*			*
Athena		+	+	+			
BirliX	*		*	*	*	*	*
Cambridge DS	*		*	*			*
Cedar		*	*	*	*	*	*
Charlotte	*		*	*			
Chorus	*		*	*			
Clouds	*		*	*		*	*
Cosmos	*		*	*	+	+	+
Cronus	*		*	*	+		
DACNOS	+		*	*		*	
DEMOS/MP	*		*	*			*
DOMAIN	*		*	*			
DUNIX	*		*	*			
Eden	*		*	*	*		*
EFS		*	*	*			
Emerald	+		*	*			
GAFFES		*	*	*	*	*	
Grapevine			*	*	*		
Guide	*		*	*		*	
Gutenberg	*		*	*	*	*	*

Name	Type of System		Transparency				
	DOS	DFS	locati- on	access	repli- cation	con- currency	failure
HARKYS		*	*	*			
HCS	+		*	*			
Helix		*	*	*			*
IBIS		*	*	*	*	*	
JASMIN	*		*				
LOCUS	*		*	*	*	*	*
Mach	*		*	*			*
Medusa	*		*	+			
Meglos	*		+	+			
MOS	*		*	*			
NCA/NCS	+		*	*	*	*	*
Newcastle		*	*	*			
NEXUS		*	*	*			*
NES		*	*	*			
Profemo	*		*	*		*	
PUISE	*		*	*	*		
QuickSilver	*		*	*			*
RES		*	*	*			
Saguaro	*		*	*	+		
S-/E-JUNIX		*		*			
SMB		+		*			
SOS	*		*	*			*
Sprite	*		*	*	*	*	
SWALLOW		*	*	*	*	*	*
V	*		*	*			
VAXcluster		*		*		*	*
XDES		*	*	*		*	*

4.2 Table of Comparison

The table of comparison is given to summarize and compare the systems discussed. It should be viewed carefully, since in certain ways any categorized comparison can be misleading. However, this way an easily legible overview may be obtained. The table provides quick access to a large amount of highly condensed information. The entries are organized according to the criteria used to describe the systems. Sometimes, a similar issue or a comparable feature for an entry has been implemented. We mark this with a special symbol (+). Here Table 1 describes the types of system and transparency issues like replication, access, withstanding failures, etc. In the comparison, Cedar, Gutenberg, NCA/NCS, Swallow performs well among all other DOS's [2] [13] [14] [16] [25] [27]. The comparison is given as follows

Table 2. Table of comparison – Hardware Requirements

criteria. Does the system provide recovery after a client or a server crash, does it support orphan detection and deletion, and is there non-volatile memory called stable storage

4.1.10 Availability

Distributed systems can be made highly available by replicating resources or services among the nodes of the network. Thus, individual indispositions of nodes can be masked. (Nested) transactions are well-suited in a computer network. Our overview covers this feature. First of all, we look at the concurrency control scheme; i. e. availability is introduced through the following mechanisms: synchronization scheme, (nested) transaction facility, and replication.

4.1.11 Process Migration

Our final point of interest is process migration. Some object-oriented systems provide mobile objects; some traditional process-based systems support migration of processes. Sometimes, these approaches come along with load-

Name	Heterogeneity	
	OS	CPUs
Accent		PERQ
Alpine	(Cedar)	Dorado
Amoeba	UNIX 4.2 BSD	68000s, PDP11, VAXes, IBM-PC, NS32016
Andrew	UNIX 4.2 BSD	SUN 2/3, MVAXes, IBMRT PC
Argus	Ulrix 1.2	MVAXes
Athena	UNIX	multi vendor HW
BirliX	UNIX 4.3 BSD	
Cambridge DS	TRIPOS	LSI-4, 68000, Z-80, PDP-11/45
Cedar		Alto, Dorado WS
Charlotte	UNIX	VAX-11/750
Chorus	UNIX V 3.2	SM 90, IBM AT PC, 68000s, 80386
Clouds	UNIX	VAX-11/750, SUN 3/60
Cosmos	UNIX	
Cronus	UNIX V7, 4.2 BSD, VMS	68000s, VAXes, SUNs, BBN C70
DACNOS	VM/CMS, PC DOS, VMS	VAXes, IBMPC, IBM370
DEMOS.MP		Z8000
DOMAIN	UNIX III, 4.2 BSD	Apollo
DUNIX	UNIX 4.1 BSD	VAXes
Eden	UNIX 4.2 BSD	SUN, VAXes
EFS	MASSCOMP RT UNIX	MASSCOMP MP
Emerald	Ulrix	MVAX II
GAFFES	all kind	all kind
Grapevine	multiple OS	multi vendor HW
Guide	UNIX V	Bull SPS 7.9, SUN 3
Gutenberg	UNIX	
HARKYS	multiple UNIX systems	multi vendor HW
HCS	multiple os	multi vendor HW
Helix	XMS	68010-based
IBIS	UNIX 4.2 BSD	VAX-11/780
JASMIN	UNIX V	
LOCUS	UNIX 4.2 BSD, V.2	DEC VAX/750, PDP-11/45, IBM PC
Mach	UNIX 4.3 BSD	all VAXes, SUN 3, IBM PC, NS32016
Medusa	StarOS	Cm*
Meglos	UNIX	68000s, VAXes, PM-68k
MOS	UNIX V7	PDP-11, PCS CADMUS 9000
NCA/NCS	DOMAINIX, UNIX 4.x BSD, VAX/VMS	SUNs, IBM PCs, VAXes
Newcastle	UNIX V7, III, BSD, Guts	PDP-11
NEXUS	UNIX 4.2 BSD	SUN
NFS	UNIX 4.2 BSD, V.2, VMS, SUN OS, Ulrix, MS/DOS	multi vendor HW
Profemo	UNIX 4.2 BSD	DEC VAXes
PULSE		LSI-11/23
QuickSilver		IBM RT-PC
RFS	UNIX V.3	multi vendor HW running UNIX V.3
Saguaro		SUN
S-F-UNIX	UNIX V	DEC PDP-11
SMB	UNIX V, UNIX 4.x BSD, VMS/DOS 3.x	multi vendor PC
SOS		
Sprite	UNIX 4.x BSD	SUN 2/3
SWALLOW		
V	UNIX 4.x BSD	SUN 2/3, VAXstation II
VAXcluster	VAX/VMS	VAX 7xx-11s, VAX/8600
XDFS		Alto

Table 2 describes the hardware requirements of various DOS and supporting version types of OS they are using. Here most of the Dos are using UNIX as their supporting OS.

Table 3 describes the changes made the different types of protocols used for the communication. The communication part includes standard, specialized protocols, shared memory and RPC based protocols. And also it compares the connection types such as VC, datagram, Pipes/Streams of the different types of DOS. In the below comparison, Cronus, Mach, performs well again all the Dos in the case of new kernel [24] [26], shared memory etc.

Table 3. Table of comparison –Kernel, Communication and Connection

Name	Changes made		Communication				Connection		
	new kernel	new layer	standard protocols	specialized protocols	shared memory	RPC-based	VC	datagram	pipes/streams
Accent	*			*				*	
Alpine	*					*			
Amoeba	*			Amoeba		*			
Andrew	*		UDP/IP			*		*	
Argus		*		ACP/IP				*	
Athena		+	TCP/IP					*	
BirliX	*				*	*		*	
Cambridge DS	*			*		*		*	*
Cedar		*	FTP					*	
Charlotte	*		*					*	*
Chorus	*		*			*		*	
Clouds	*				*	*			
Cosmos	*								
Cronus	*		TCP, UDP	VLN		*	*	*	*
DACNOS		*	OSI			*		*	
DEMOS.MP	*			*		*		*	
DOMAIN	*			*	*				
DUNIX	*		TCP/IP	*				*	
Eden		*				*			
EFS	*			RDP		*			
Emerald		*							
GAFFES		+	*			*	*		
Grapevine			*					*	*
Guide		*			*				
Gutenberg		*				*			
HARKYS		*	*			*			
HCS		*	*			*			
HARKYS		*	*			*			
HCS		*	*			*			
Helix	*		OSI			*		*	
IBIS		*	TCP/IP			*	*	*	
JASMIN	*			Paths					
LOCUS	*					*			
Mach	*			*	*	*	*	*	*
Medusa	*				+				*
Meglos	*			*		*		*	
MOS	*		UDP/IP			*		*	
NCA/NCS	*		UDP/IP			*		*	
Newcastle	*					*			
NEXUS	*	*	*			*		*	
NFS	*		UDP/IP			*		*	
Profemo	*	*	UDP/IP			*	*	*	*
PULSE	*					*	*	*	
QuickSilver	*			*		*	*	*	
RFS	*		OSI TLI						*
Saguaro									*
S-F-UNIX	*			KP			*		
SMB	*			SMB			*		
SOS	*			*		*	*	*	
Sprite	*			*		*			
SWALLOW		*		SMP				*	
V	*			VMTP		*	*	*	
VAXcluster	*			MSCP		*	*	*	
XDFS		*		Pup				*	

Table 4 describes the issues like semantics, naming and security. In the below comparison Amoeba, GAFFES, and Alpine performs well especially in the object oriented and Encryption related things.

Table 4. Table of comparison – Semantics, Naming and Security

Name	Semantics			Naming		Security			
	may be	at most once	exactly once	object-oriented	hier-archial	en-cryption	special HW	capa-bilities	mutual auth.
Accent		*		*				*	*
Alpine		*			*	*			*
Amoeba		*		*		*	*	*	
Andrew	*				*				*
Argus			*	*					
Athena					*				
BirliX	*			*	*			*	
Cambridge DS	*	*		*				*	
Cedar		*			*				
Charlotte	*				*			*	
Chorus				*					
Clouds			*	*					
Cosmos		*		*				+	
Cronus	*			*				+	
DACNOS		*		*					*
DEMOS/MP	*		*		*			+	
DOMAIN				*	*				
DUNIX	*				*				
Eden		*		*				*	
EFS	*				*				
Emerald	*			*					
GAFFES			*		*	*	*	*	*
Grapevine					*				
Guide		*		*					
Gutenberg			*	*				*	
HARKYS	*				*				
HCS									
Helix				*		*		*	
IBIS	*				*				*
JASMIN	*				*			*	
LOCUS		*	*		*				
Mach		*		*				*	
Medusa				*					
Meglos					*				
MOS	*				*				
NCA/NCS		*		*					
Newcastle	*	*			*				
NEXUS	*		*	*	*				
NFS	*				*	*			
Profemo				*				*	
PULSE				*					
QuickSilver		*	*		*				
RFS	*				*				
Saguaro					*				
S-F-UNIX	*				*				
SMB	*				*				
SOS	*	*		*				*	
Sprite		*			*				
SWALLOW		*		*				*	
V		*			*				*
VAXcluster	*				*				
XDFS		*			*				

Name	Availability				Failures				
	synchro-nization	TA	nested TA	repli-cation	recovery client crash	recovery server crash	stable storage	orphan detection	Process Migration
Accent					+	+			*
Alpine	+	*			+	+			
Amoeba	*					*	*	*	
Andrew	*			*					
Argus	*	*	*	*	*	*	*	*	
Athena									
BirliX	*			*	*	*		*	*
Cambridge DS		+				*		+	
Cedar	*	*		*	*	*			
Charlotte						*			*
Chorus									*
Clouds	*	*	*		*	*	*		
Cosmos	*	*		*					
Cronus				*					*
DACNOS	*								
DEMOS/MP					*	*			*
DOMAIN									
DUNIX									
Eden		*		*		*			*
EFS		*			*	*			
Emerald	+								*
GAFFES	*	*	*	*					
Grapevine				*					
Guide	+	*	*						
Gutenberg	*	*	*	*			*		
HARKYS									
HCS									
Helix		*	*		*			+	
IBIS	*			*	*	*			
JASMIN									
LOCUS	*	*	*	*	*	*			*
Mach					+	+			*
Medusa	*								*
Meglos	+								
MOS				*					*
NCA/NCS	+			*					
Newcastle									
NEXUS		*	*						
NFS					*	*			
Profemo	*	*	*						
PULSE	*			*					
QuickSilver		*				*	*		
RFS	*				*				
Saguaro				*					
S-F-UNIX									
SMB									
SOS					*	*			
Sprite	*			*					*
SWALLOW	*	*		*	*	*	*		
V									*
VAXcluster	*				*	*			
XDFS	*	*			*	*	*		

Table 5 describes about the comparison of availability issues dealing with synchronization, Replication and the issues regarding failures such as, recovery client crash, recovery server crash, stable storage, orphan detection etc. In the below comparison, Amoeba, Argus, Cedar, Locus, Swallow, XDFS performs well among all the things specially in the issues like replication, recovery server crash, process migration.

Table 5. Table of comparison – Availability, Failures

5. Summary

Distributed operating systems are still in an early phase of development, with many unanswered questions and relatively little agreement among workers in the field about how things should be done. Many experimental systems use the client-server model with some form of remote procedure call as communication base, but there are also systems built on the connection model. Relatively little has been done on distributed naming, protection, and resource management, other than building straight-forward name servers and process servers. Fault tolerance is an up-and-coming area, with work progressing in redundancy techniques and atomic actions. Finally, a considerable amount of work has gone into the construction of file servers, print servers, and

various other servers, but here too there is much work to be done. The only conclusion that we draw is that distributed operating systems will be an interesting and fruitful area of research for a number of years to come.

References

- [1] Adams, C. J., Adams, G. C., Waters, A. G., Leslie, I., and Kirk, P., "Protocol Architecture of the Universe Project," In Proceedings of the 6th International Conference on Computer Communication (London, Sept. 7-10). International Conference for Computer Communication, pp. 379- 383, 2001.
- [2] Almes, G. T., Black, A. P., Lazowska, E. D., and Ni! Ie, J. D., "The Eden System: A Technical Review," IEEE Trans. Softw. Eng. Se-11 (Jan.) 43-59, 2006.
- [3] Anderson, T., and Lee, P. A. Fault Tolerance, "Principles And Practice," Prentice-Hall International, London, 2000.
- [4] Avizienis, A., and Chen, L., "On the Implementation of N-Version Programming for Software Fault-Tolerance During Execution," In Proceedings of the International Computer Software and Applications Conference. IEEE, New York, pp. 149-155, 2008.
- [5] Avizienis, A., and Kelly, J., "Fault Tolerance by Design Diversity," Computer 17 (Aug.), 66-80, 1984.
- [6] Bal, H. E., Van Renesse, R., and Tanenbaum, A. S., "A Distributed, Parallel, Fault Tolerant Computing System," Rep. 1%106, Dept. of Mathematics and Computer Science. Vriie Univ., The Netherlands, Oct. 1999.
- [7] Ball, J. E., Feldman, J., Low, R., Rashid, R., and Rovner, P., Rig, "Rochester's Intelligent Gateway: System Overview," IEEE Trans. Softw. Eng. Se-Z (Dec.), 321-329.
- [8] Barak, A., and Shiloh, A. A., "Distributed Load-Balancing Policy for a Multicomputer," Softw. Pratt. Expert. 1.5 (Sept.), 901-913, 1985.
- [9] Birman, K. P., and Rowe, L. A., "A Local Network Based on the Unix Operating System," IEEE Trans. Softw. Eng. Se-8 (Mar.), 137-146, 1982.
- [10] Birrell, A. D., "Secure Communication Using Remote Procedure Calls," ACM Trans. Compute. Syst. 3, 1 (Feb.), 1-14, 1985.
- [11] Birrell, A. D., and Needham, R. M., "A Universal File Server," IEEE Trans. Softw. Eng. Se-6, (Sept.), 450-453, 1980.
- [12] Birrell, A. D., and Nelson, B. J., "Implementing Remote Procedure Calls," ACM Trans. Compute. Syst. 2, 1 (Feb.), 39-59, 1984.
- [13] Birrell, A. D., Levin, R., Needham, R. M., and Schroeder, M., "Grapevine: An Exercise in Distributed Computing," Commun. Acm 25, 4 (Apr.), 260-274, 1982.
- [14] Birrell, A. D., Levin, R., Needham, R. M., and Schroeder, M., "Experience with Grape-Vine: The Growth of a Distributed System. ACM Trans. Compute. Syst. 2, 1 (Feb.), 3-23, 1984.
- [15] Black, A. P. "An Asymmetric Stream Communications System," Oper. Syst. Rev. (ACM) 17, 5, 4-10, 1983.
- [16] Black, A. P., "Supporting Distributed Applications: Experience with Eden," In Proceedings of the 10th Symposium on Operating Systems Principles (Orcas Island, Wash., Dec. L-4). ACM, New York, pp. 181-193, 1985.
- [17] Borg, A., Baumbach, J., and Glazer, S., "A Message System Supporting Fault Tolerance," Oper.Syst. Rev. (ACM) 17, 5, 90-99, 1983.
- [18] Brown, M. R., Kolling, K. N. and Tag. E. A., "The Alnine File System," ACM Trans. Com- Put. Syst. 3, 4 ~Nov.), 261-293, 1985.
- [19] Brownbridge, D. R., Marshall, L. F., Andrandell, B., "The Newcastle Connection-or Unixes of the World Unite! Softw," Pratt. Expert. 12 (Dec.), 1147-1162, 1982.
- [20] Bryant, R. M., and Finkel, R. A., "A Stable Distributed Scheduling Algorithm," In Proceedings of the 2nd International Conference on Distributed Computer Systems (Apr.). IEEE, New York, pp. 314-323, 1981.
- [21] Chandy, K. M., Misra, J., and Haas, L. M., "Distributed Deadlock Detection," ACM Trans. Compute. Syst. 1, 2 (May), 145-156, 1983.
- [22] Cheriton, D. R.M, "The Thoth System: Multi- Process Structuring and Portability," American Elsevier, New York, 1982.
- [23] Cheriton, D. R., "An Experiment Using Registers for Fast Message-Based Inter process Communication," Oper. Syst. Rev. 18 (Oct.), 12-20, 1984a.
- [24] Cheriton, D. R., "The V Kernel: A Software Base for Distributed Systems," IEEE Softw. 1 (Apr.), 19-42, 1984b.
- [25] Cheriton, D. R., and Mann, T. P., "Uniform Access to Distributed Name Interpretation in the V System," In Proceedings of The 4th International ConferenceOn Distributed Computing Systems. IEEE, New York, pp. 290-297, 1984.
- [26] Cheriton, D. R., and Zwaenepoel, W., "The Distributed V Kernel and its Performance or Disk- Less Workstations," In Proceedings of the 9th Sym- Podium on Operating System Principles. ACM, New York, pp. 128-140, 1983.
- [27] Cheriton, D. R., and Zwaenepoel, W., "One-to-Many Interprocess Communication in the V-System. In Szgcomm," '84 Tutorials and Svmposkm on Communications Architectures and Protocols (Montreal, Quebec, June 6-8). ACM, New York, 1984.
- [28] Cheriton, D. R., Malcolm, M. A., Melen, L. S., and Sager, G. R. Thoth, "A Portable Real- Time Operating System," Commun. ACM 22, 2 (Feb.), 105-115, 1979.
- [29] Chesson, G., "The Network Unix System," In Proceedings of The 5th Symposium on Operating Systems Principles," (Austin, Tex., Nov. 19-21). ACM, New York, pp. 60-66, 1975.
- [30] Chow, T. C. K., and Abraham, J. A., "Load Balancing in Distributed Systems," IEEE Trans. Softw. Eng. Se-8 (July), 401-412, 1982.
- [31] Chow, Y. C., and Kohler, W. H., "Models for Dynamic Load Balancing in Heterogeneous Multiplex Processor Systems," IEEE Trans. Compute. C-28 (May), 354-361, 1979.

- [32] Chu, W. W., Holloway, L. J., Min-Tsung, L., and Efe, K., "Task Allocation in Distributed Data Processing," *Computer* 23 (Nov.), 57-69, 1980.
- [33] Curtis, R. S., and Wi~Ie, L. D., Global Naming In Distributed Systems. *IEEE Softw.* 1, 76-80, 1984.
- [34] Dalal, Y. K., "Broadcast Protocols in Packet Switched Computer Networks," Ph.D. Dissertation, Computer Science Dept., Stanford Univ., Stan- Ford, Calif.
- [35] Dellar, C., " A File Server for a Network of Low-Cost Personal Microcomputers," *Softw. Pratt. Erper.* 22 (Nov.), 1051-1068, 2009.
- [36] Dennis, J. B., and Van Horn, E. C., "Programming Semantics for Multiprogrammed Computations. *Commun.*" *ACM* 9, 3 (Mar.), 143- 154, 2009.
- [37] Dion, J, "The Cambridge File Server," *Oper. Syst. Reu. (ACM)* 14 (Oct.), 41-49.
- [38] Efe, K., "Heuristic Models of Task Assignment Scheduling in Distributed Systems," *Computer* 15 (June), 50-56, 1992.
- [39]Eswaran, K. P., Gray, J. N., Lorie, J. N., and Traiger, I. L., "The Notions of Consistency and Predicate Locks in a Database System," *Com- Mun. ACM* 19, 11 (Nov.), 624-633, 1986.
- [40]Farber, D. J., and Larson, K. C., "The System Architecture of the Distributed Computer System-The Communications System," In *Proceedings of the Symposium on Computer Networks (Brooklyn, Apr.)*. Polytechnic Inst. of Brooklyn, Brooklyn, N.Y, 1972.

Mapping and Generation Model for Named Entity Transliteration in CLIR Systems

V. Narasimhulu, P. Sujatha and P. Dhavachelvan

Department of Computer Science, Pondicherry Central University,
Pondicherry - 605014, India.
{narasimhavasi, spothula, dhavachelvan}@gmail.com

Abstract: Named Entities are the expressions in human languages that explicitly link notations to the entities in the real world. They play an important role in Cross Lingual Information Retrieval (CLIR), because most of the user queries contain Out Of Vocabulary (OOV) terms with majority of named entities. Missing their translations has a significant impact on the retrieval system. Translation of named entities can be done using either named entity translation or named entity transliteration. Named entity translation causes translation failures, since if a given name is not found or new to the translation system, it may be discarded or mistranslation occurs. Transliteration is the suitable method for translating named entities and is a challenging task. This paper, discusses various transliteration techniques and a problem in the Compressed Word Format (CWF) mapping model. A system is proposed based on mapping and generation models. The proposed system follows one of the major techniques of transliteration called grapheme-based transliteration model. It transliterates all the source language names which are specified by the user and gives the right equivalent and relevant target names.

Keywords: Named entities, cross lingual information retrieval, translation, transliteration, grapheme.

1. Introduction

CLIR is the process of submitting query in source language and retrieving information in target language. This processing requires mainly three phases: Text processing, query translation and retrieval system. Text processing includes morphological analyzer, stop words removal and stemming process. Morphological analyzer analyzes the structure of the words in a given query. E.g. verbs, adverbs, adjectives etc. Stop words removal removes the stop words in a given query. E.g. the, is, was, can, should etc. Stemming is the process of reducing inflected words to their base or root form. E.g. fishing, fished and fisher are reduced into the root word fish. After the text processing, the source query contains a combination of dictionary and OOV words. Translation of dictionary words can be done using bilingual dictionary. Translation of OOV terms cannot be handled using bilingual dictionary because of its limited coverage. OOV words are significant for retrieving information in a CLIR system. The retrieval effectiveness can reduce up to 60% if OOV terms are not handled properly [1].

OOV terms can be of many types; some of them newly formed words, loan words, abbreviations or domain specific terms, but the biggest group of OOV terms, which are observed to be, as many as half of the whole observed OOV terms in [1], belongs to a group called named entities.

Named entities are the expressions in human languages that explicitly link notations in languages to the entities in the real world. Examples of named entities are individual name, role, organization, location, date, time, quantity, numerical expression, phone number etc.

Named entities form a very dynamic set, already there exists a large quantity of them, and at the same time people are creating new named entities every day. This makes that dictionary cannot cover all named entities. They play important role in locating relevant documents. The occurrence of named entities in user queries makes easier for retrieval system, if the correct translations of named entities are available [2]. The retrieval performance or average precision of CLIR reduces significantly, when named entities are not translated properly in queries [3]. Translation of named entities can be done using named entity translator, which causes translation failures. i.e. either drops the word or mistranslates, if the given name is new to the translation system. Another possibility for translation of named entities is named entity transliteration. Previous studies [4] show that the average precision score of a CLIR system get reduced by 50% when the named entities were not properly transliterated. Therefore transliteration of named entities from source language to the target language presents an interesting challenge.

Transliteration is the process of transforming a word written in a source language into a word in a target language without the aid of a resource like a bilingual dictionary. It refers to expressing a word in one language using the orthography of another language. Orthography means the art or study of correct spelling according to established usage. Transliteration can be classified into two directions: forward transliteration and backward transliteration. Given a pair (s, t), where s is the source word in source language and t is the transliterated word in target language. Forward transliteration is the process of converting s into t. Backward or back transliteration is the process to correctly find or generate s for a given t. This paper is used in the work of forward transliteration.

The major techniques for transliteration can be classified into three categories: grapheme-based, phoneme-based and Hybrid transliteration models [5].

Grapheme refers to the basic unit of written language or smallest contrastive units. In grapheme-based transliteration spelling of the original string is considered as a basis for transliteration. It is referred to as the direct method because it directly transforms source language graphemes into target language graphemes without any phonetic knowledge of the

source language words. Here transliteration is identified by mapping the source language names to their equivalent names in a target language and generating them.

Phoneme refers to the simplest significant unit of sound or the smallest contrastive units of a spoken language. In phoneme-based transliteration pronunciation rather than spelling of the original string is considered as a basis for transliteration. Phoneme based transliteration is referred as a pivot method because it uses source language phonemes as a pivot, when it produces target language graphemes from source language graphemes. It usually needs two steps:

- Produce source language phonemes from source language graphemes.
- Produce target language graphemes from source phonemes.

These two steps are explicit if the transliteration system produces target language transliterations after producing the pronunciations of the source language words; they are implicit if the system uses phonemes implicitly in the transliteration stage and explicitly in the learning stage [6]. ARPAbet symbols are used to represent source phonemes. ARPAbet is one of the methods used for coding source phonemes into ASCII characters [7]. It is developed by Advanced Research Projects Agency (ARPA) as part of their Speech Understanding Project. ARPAbet symbols for English consonants are given in Table 1, and for English vowels are given in Table 2.

Table 1. Arpabet symbols for English consonants

P	T	K	B	D	G	M	N	NG	F
V	TH	DH	S	Z	SH	ZH	CH	JH	L
W	R	Y	H	Q	DX	NX	EL		

Table 2. Arpabet symbols for English vowels

IY	IH	EY	EH	AE	AA	AO	UH	OW	UW
AH	ER	AR	AW	OY	AX	AXR	IX	UX	

Grapheme-based and phoneme-based transliteration is referred to as hybrid transliteration. It makes use of both source language graphemes and phonemes, to produce target language transliterations. Here after, a source language grapheme is a source grapheme, a source language phoneme is a source phoneme, and a target language grapheme is a target grapheme.

In each model, transliteration of a source language to target language is an interesting and challenging task. The present paper discusses different issues and problems regarding transliteration from source language to target language and also problem in the CWF mapping model. The present paper proposes a system based on grapheme-based transliteration model.

The organization of the paper is as follows. In Section 2, the related works on different transliteration models and transliteration of named entities are described. Description of CWF mapping model, problem in the CWF mapping model

and problem definition are given in the same Section. The description of the proposed system and comparison of proposed system with existing systems is given in the same Section 3. Section 4, describes about implementation details of the mapping and generation model. The conclusion of the paper is given in Section 4.

2. Related Work

Transliteration is the process of transcribing words from the source script to a target script. Grapheme-based transliteration considers the spelling or characters of the original string as the basis for transliteration. Previous studies have proposed several methods with this model. An n-gram based statistical transliteration model for English to Arabic names was described in [4]. It presents a simple statistical technique, which does not require any heuristics or linguistic knowledge of either language. It is specified that transliteration either of OOV named entities or of all OOV words is an effective approach for CLIR. A decision tree based transliteration model [8], is a language independent methodology for English to Korean transliteration and supports back transliteration. It is composed of character alignment and decision tree learning. Transliteration and back transliteration rules are induced for each English alphabet and each Korean alphabet. A maximum entropy based model [9] is an automatic transliteration model from English to Japanese words and it successfully transliterates an English word not registered in any bilingual or pronunciation dictionaries by converting each partial letters in the English word into Japanese characters. A new substring based transliteration method based on phrase-based models of machine translation was described in [10]. Substring based transliteration method is applied on Arabic to English words.

Phoneme-based transliteration considers the pronunciation of the word as the basis for transliteration. Previous studies have proposed several methods with this model. A rule based model for English to Korean transliteration using pronunciation and context rules is described in [11]. It uses phonetic information such as phoneme and its context as well as orthography of English language as the basis for transliteration. A machine-learned phonetic similarity model [12] is a backward transliteration model and provides learning algorithm to automatically acquire phonetic similarities from a corpus. Given a transliterated word, similarity based model compares the list of source candidate words and the one with highest similarity will be chosen as the original word. The first semantic transliteration of individual names was given in [13]. It is a phoneme-based transliteration and based on the word's original semantic attributes. It is a probabilistic model for transliterating person names in Latin script into Chinese script. Proved semantic transliteration substantially improves accuracy over phonetic transliteration.

Hybrid transliteration is a combination of both grapheme-based and phoneme-based transliteration. Oh and Choi [5] proposed a model for improving machine transliteration using an ensemble of three different transliteration models for English to Korean and English to Japanese languages. Three transliteration models are grapheme, phoneme and both. Bilac and Tanaka [14] proposed a new hybrid back

transliteration system for Japanese, which contains segmentation, phoneme-based and grapheme-based transliteration modules. The system first finds the best segmentation of transliterated string and then obtains back transliteration using the combined information based on pronunciation and spelling. Hong et al. [15] proposed a hybrid approach to English-Korean name transliteration. It is based on phrase-base Statistical Machine Translation (SMT) model with enabled factored translation features. The system is combined with various transliteration methods including a Web-based n-best re-ranking, a dictionary-based method and a rule-based method.

In the Indian language context, transliteration similarity mechanism to align English-Hindi texts at the sentence and word level in parallel corpora was given by [16]. This is based on a grapheme-based model. It describes a simple sentence length approach to perform sentence alignment and multi feature approach to perform word alignment. Punjabi machine transliteration was given by [17]. It addresses the problem of transliteration for Punjabi language from Shahmukhi (Arabic script) to Gurmukhi using a set of transliteration rules (character mappings and dependency rules). A discriminative, Conditional Random Field (CRF)-Hidden Markov Model (HMM) model for transliterating words from Hindi to English was used in [18]. The model is a statistical transliteration model, which generates desired number of transliterations for a given source word. It is based on grapheme-based model and language independent. A word origin based transliteration for splitting Indian and foreign origin words based on their phoneme equivalents was shown by [19]. The given transliteration mechanism is applicable for Indian languages and shown that word origin is an important factor in achieving higher accuracy in transliteration. A phrase or grapheme-based statistical machine transliteration of named entities from English to Hindi using a small set of training and development data was shown by [20]. A CWF mapping model for transliterating named entities from English to Tamil was given by [21]. This is based on grapheme-based model in which transliteration equivalents are identified by mapping the source language names to their equivalents in target language database.

2.1 Named Entity Recognition (NER)

NER is the subtask of the information extraction that seeks to locate and classify atomic elements in a text into predefined categories. It extracts the specific information from the text or document. It has important significance in internet search engines and performs important tasks in many of the language engineering applications such as machine translation, Question-Answering (QA) systems, indexing for information retrieval and automatic summarization. Named entity recognizer is language dependent. Each language needs a separate named entity recognizer. The following shows a simple example for recognizing named entities in a text using named entity recognizer or tagger.

E.g. Ram bought 3000 shares.
 <ENAMEX TYPE="PERSON">Ram</ENAMEX> bought
 <ENAMEX TYPE="QUANTITY">3000</ENMAEX>
 shares.

In the above example the annotations have been done using so called ENAMEX tags. A large number of techniques have been developed to recognize named entities for different languages. Some techniques are rule based and others are statistical or machine learning techniques [22], [24]. These techniques are summarized in Table 3.

Table 3. NER Techniques

S. No	Name of the Technique	Description	
1	Rule Based Technique	It uses the morphological and contextual evidence of a natural language and a consequently determines the named entities.	
2	Statistical Learning Techniques	Supervised Learning	In this NER process is learned automatically on large text corpora and the supervised by a human.
		Unsupervised Learning	In this NER process is not supervised, instead existing semantic lexical databases such as WordNet are consulted automatically
		Semi-supervised Learning	It involves a small degree of supervision, such as a set of seeds, for learning the process.

2.2 CWF Mapping Model

CWF mapping model [21] is a grapheme based transliteration model, where transliteration equivalents are identified by mapping the source language names to their target language database, instead of generating them. The basic principle is to compress the source word into its minimal form and align it across an indexed list of target language words to arrive at the top n-equivalents based on the edit distance. That is, for a given a source language named entity (English) string, it will produces a ranked list of transliterated names in the target language (Tamil).

In this model individual names in the target language are collected manually and listed in the database by running named entity recognizer on the archives. These names are then romanized so that they can be easily compared to the English queries. For a given English named entity, compress both English named entity and list of collected Tamil names into minimal consonant form based on a set of linguistic rules. Linguistic rules are an ordered set of rewrite and remove rules. Rewrite rules replace characters or clusters of characters with other characters or clusters. Remove rules simply remove the characters or clusters of characters. For mapping Tamil names, custom Romanization scheme was used. This scheme maps every Tamil character to Roman characters in a one-to-one mapping fashion. One-to-one fashion means each Tamil letter is considered as a single roman character. After compressing both source and list of target names in the database index, right equivalents were matched using the Modified Levenshtein algorithm [21], by calculating edit distance between source name and list of target names in the database index. Here mapping is done between compressed word of a source name and compressed word of a target name. Levenshtein edit distance is normally

used for finding number of changes, between two strings of same language and contains insertion, deletion or substitution of a single character [23]. Modified Levenshtein is a variant of Levenshtein algorithm and modified based on considering character sets of source and target languages, since source and target language strings use different character sets. The edit distance between perfect matching pairs can be zero. The algorithm has been proved to be effective for matching perfect pairs. Instead of single perfect matching pair, CWF mapping model retrieve n-equivalent matching pairs based on the nearest edit distance value for CLIR applications.

2.2.1. Advantages of CWF mapping model

- CWF mapping model is more accurate than the statistical generation models in English-to-Tamil transliteration.
- In case of mapping based approaches, CWF is more accurate and precise than the actual word forms.
- Using CWF, half of the execution time is reduced. The reason is mapping is done between the compression words, not between the actual words.

2.3 Problem in CWF Mapping Model

CWF mapping model accurately map the right matching pair between source and target languages, only if a given source named entity has a right matching equivalent in the target language database index. Mapping is done between the compressed word of source name and compressed word of target name. Suppose a given source named entity does not have a right equivalent matching pair in the target language database, it gives n-relevant matching pairs based on the edit distance, not the required exact equivalent matching word, because, the previous model [21] would not be generating the target transliterations and not updating the database index regularly. Except collecting and listing the target language names in the database. The drawback of the system is: mapping is taking place only with the source names which are presented in the database. It would not work for other source names which are not in the database.

2.4 Problem Definition

To overcome the problem, a system is proposed based on mapping and generation. It is a grapheme-based transliteration model, which generates or transliterates the source name into equivalent target name, if the right equivalent is not available in database. After transliterating, web is used to retrieve relevant words for finding relevant equivalents and database is updated regularly

3. Proposed System

3.1 Introduction

A system is proposed for transliterating named entities from source to target language, which is based on major technique of transliteration called grapheme-based transliteration. The proposed system includes mapping and generation models. It transliterates given source name into equivalent target name

and retrieves relevant words. This equivalent and relevant words are displayed to the user.

3.2 Proposed System Model

The overview of the proposed system is shown in Figure 1. It mainly composed with the following modules: Word Compression, Mapping, Transliteration, Web Retrieval and Updation. The working principle of individual modules can be described in the following paragraphs.

3.2.1 Word Compression Module

For a given source language name (SN), this module compress as it into a minimum consonant skeleton form or CWF form based on a set of linguistic rules. Linguistic rules are an ordered set of, combination of rewrite and remove rules as specified in [21]. The output of this module is named as compressed source language name (CSN). The target language names $\{TN_1, TN_2 \dots TN_n\}$ in the database are compressed in the same manner by using the linguistic rules, but their rule sets are different. Target language database index is in the form of tuples, where each tuple contains both compressed name and actual name.

3.2.2 Mapping Model

This model receives CSN as input and searches all compressed target names $\{CTN_1, CTN_2 \dots CTN_n\}$ from the database index. It converts CSN and $\{CTN_1, CTN_2 \dots CTN_n\}$ into intermediate scheme for finding exact equivalent compressed target name (CETN). Intermediate scheme acts as a mediator for mapping between source and target languages because both have different character representations. Roman scheme is the intermediate scheme developed for mapping or aligning characters between source and target languages. A scheme having one-to-one configuration is used here for mapping between CSN and $\{CTN_1, CTN_2 \dots CTN_n\}$. The model then calculates edit distance between compressed source name and each compressed target names $\{ED(CSN, CTN_1), (ED(CSN, CTN_2) \dots (CSN, CTN_n)\}$.

Calculate $\{ED(CSN, CTN_1), (ED(CSN, CTN_2) \dots (CSN, CTN_n)\} = \{ED_1, ED_2 \dots ED_n\}$.

Modified Levenshtein algorithm is used for finding edit distances between CSN and $\{CSN_1, CSN_2 \dots CSN_n\}$ as described in the paper [21]. After finding $\{ED_1, ED_2 \dots ED_n\}$, each edit distance is checked with equivalent to zero for finding exact target equivalent. If one of $\{ED_1, ED_2 \dots ED_n\}$ is equivalent to zero, then CETN is found. Therefore, the corresponding actual target name (ETN) and relevant actual target names $\{TR_1, TR_2 \dots TR_m\}$ are retrieved from the database based on the minimum edit distance value. Finally ETN and $\{TR_1, TR_2 \dots TR_m\}$ are displayed in the user interface. When there is more than one candidate at the same edit distance, finer ranking can be made based on the edit distance between the actual forms of source and target strings. There is no chance for two candidates having zero edit distance, because one string can have only one equivalent not more than one. Suppose none of $\{ED_1, ED_2 \dots ED_n\}$ is equivalent to zero, exact ETN is not found

S. No	Existing System (CWF Mapping Model)	Proposed System
1	Transliteration is based on only database target names.	Transliteration is based on other than database target names.
2	Grapheme based mapping model.	Grapheme based mapping and generation model.
3	Specifies top-n equivalents with relevance ranking.	Specifies top-n equivalents with relevance ranking.
4	Database is constructed manually and need not updated regularly.	Database is updated with the generated target name and relevant words in the compressed and actual forms.
5	Precision is less, if the given source name does not have equivalent target name in the database.	Precision is good because it is transliterating and giving the equivalent target name, if it is not in the database.
6	No exact equivalency for other than database names	Exact equivalency is good for all the source names.

Precision is one of the performance measures for retrieval systems. It specifies the measure of exactness. Here, it is used to measure the exact equivalence with respect to the source name. CWF mapping model has more precision on only database names, where as proposed system probably have good precision on all the names other than database. CRF-HMM model does not reduce the edit distance between source and target names, where as proposed system reduce.

Table 5. Comparison of characteristics with CRF-HMM model and CWF mapping model

S. No	Characteristics	CRF-HMM Model	CWF Mapping Model	Proposed System
1	Generation model	Yes	No	Yes
2	Compression mapping model	No	Yes	Yes
3	Supports all the source names	Yes	No (Only database)	Yes
4	Specifies top-n equivalents	Yes	Yes	Yes
5	Transliteration accuracy	Less	More (Only for database names)	More (For all the names)
6	Execution time	More	Less	Less for database names and more for other names
7	Exact Equivalence	Good (For all the names)	More (Only for database names)	Good (For all the names)
8	Reduce edit distance between source and target names	No	Yes (on compressed names)	Yes (on compressed names)

edit distance because of Modified Levenshtein algorithm and compressing the source and target names

4. Implementation and Evaluation

4.1 Implementation Details

This section describes the implementation of the mapping and generation model. It is implemented using GUI (Graphical User Interface) components of the Java

programming. MSAccess is used as a database for maintaining target language names. Unicode is used for storing the target names in the database. Unicode is a standard which provides a unique number for every character, no matter what the language is. Here, English language is used as a source language for receiving input from the user and Telugu language is used as a target language to display the output.

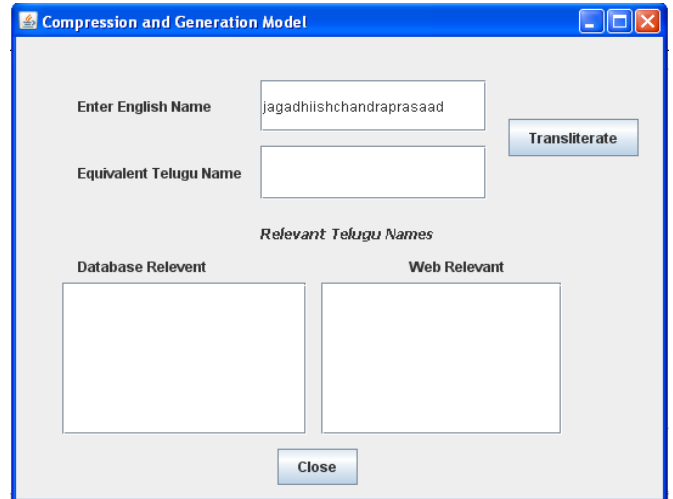


Figure 2. Source name specified by the user

For a given English name as shown in Figure 2, mapping and generation model retrieves the exact equivalent and relevant Telugu names, if the given English name has exact equivalent Telugu name in the database and displays at the user interface as shown in Figure 3. The proposed model displays topmost 5 relevant Telugu names at the user interface and set the minimum edit distance is less than equal to two (≤ 2). Otherwise mapping and generation model transliterates the English name into equivalent Telugu name and retrieves the minimum edit relevant Telugu names (≤ 2) from the web and stored in a file. Finally, equivalent and relevant Telugu names are given to the user interface as shown in Figure 4, and they are updated in the database.

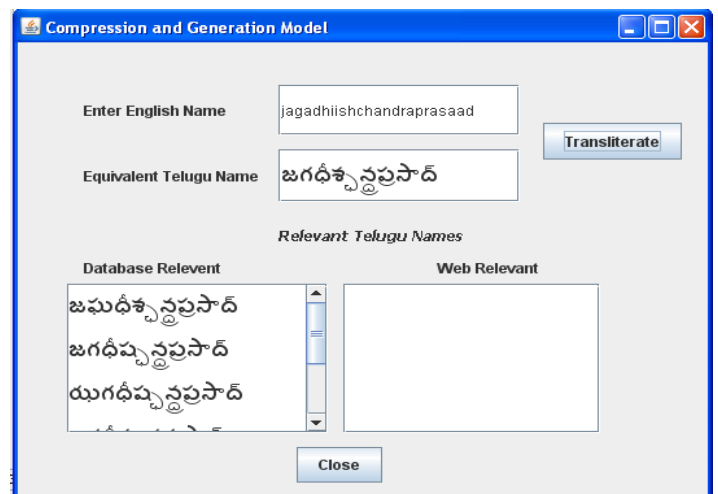


Figure 3. Equivalent and relevant target names from the database

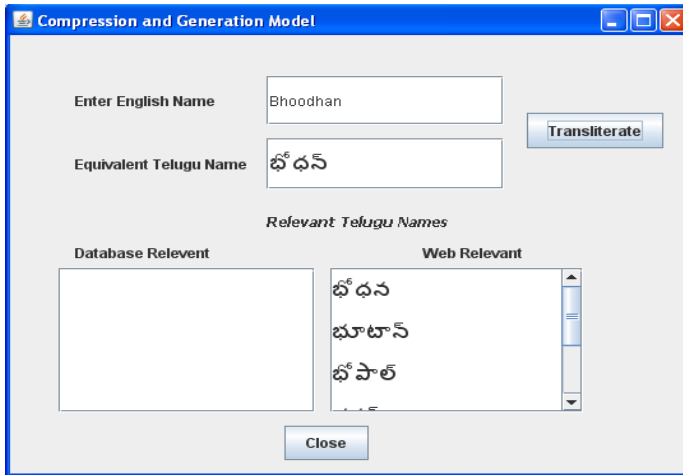


Figure 4. Equivalent transliterated target name and relevant target names from the web

4.2 Evaluation Results

To evaluate performance and accuracy of the compression and generation model stored 800 Telugu named entities in the database. The list of different categories of named entities stored in the database is specified in Table 6.

Table 6. Categories of named entities stored

Category	Number of Entities
Person names	410
Place names	224
Organization names	166

Database is designed with both compressed and actual names of above category named entities. Totally 100 English named entities were used to verify the proposed system. In that, 50 named entities have equivalent Telugu names in the database. This is for evaluating the system whether it is correctly identifying the equivalent name and retrieving the relevant names. Remaining English named entities were used whether it is correctly transliterating into equivalent Telugu names and retrieving relevant names from the web. After above evaluation process, all the relevant Telugu names received from the web are updated in the database. From the above evaluation performance, equivalency and transliteration accuracy is calculated and given in Table 7.

Table 7. Evaluation of the proposed system

English Named Entities (100)	Performance	Equivalency	Transliteration Accuracy
Equivalent English named entities (50)	97.60%	98.74%	97.62%
Non-Equivalent English named entities (50)	95.96%	98.36%	97.04%

The above results clearly shows that the compression and generation model can be used for both equivalent and non equivalent named entities in the database.

5. Conclusion and Future Work

Named Entities are the expressions in human languages that explicitly link notations to the entities in the real world. They play an important role in CLIR. Transliteration is the relevant technique for translating named entities between source and target languages. All the major techniques and general problems in the transliteration are described in this paper. A system is proposed based on grapheme-based transliteration, which includes mapping and generation. It reduces the search processing time in the database because of clustered index. It improves transliteration accuracy and equivalency of the system by generating the target name and retrieving relevant words from the web effectively and efficiently.

Our current work can be extended further by integrating proposed model with the CLIR system and reducing the search process in the database index. An interesting future scope is going to use effective string matching methods for matching between source and target names. Finally, this work can be extended on more than one target language. That is Multi Lingual Information Retrieval (MLIR).

References

- [1] D. Demner-Fushman and D. W. Oard, "The effect of bilingual term list size on dictionary-based cross-language information retrieval," *In 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, pp. 108-118, 2003.
- [2] T. Mandl and C. Womser-Hacker, "The effect of named entities on effectiveness in cross-language information retrieval evaluation," *In ACM SAC'05*, pp. 1059-1064, 2005.
- [3] L. S. Larkey, N. A. Jaleel, and M. Connell, "What's in a Name?: Proper names in arabic cross language information retrieval," *CIIR Technical Report, IR-278*, 2003.
- [4] N. A. Jaleel and L. S. Larkey, "Statistical transliteration for english-arabic cross language information retrieval," *In Proceedings of the twelfth international conference on Information and knowledge management*, November 03-08, 2003, New Orleans, LA, USA.
- [5] J. H. Oh and K. S. Choi, "An ensemble of transliteration models for information retrieval," *Information Processing and Management: an International Journal*, v.42 n.4, pp. 980-1002, July 2006.
- [6] S. Bilac and H. Tanaka, "Improving back-transliteration by combining information sources," *In Proceedings of IJC-NLP*, pp. 542-547, 2003.
- [7] D. Jurafsky and J. H. Martin, "Speech and Language processing: An introduction to natural language processing," *Computational Linguistics and Speech Recognition*, 2007.
- [8] B. J. Kang and K. S. Choi, "Automatic transliteration and back- transliteration by decision tree learning," *In:*

Proc. Of the Second International Conference on Language Resources and Evaluation, 2000.

[9] I. Goto, N. Kato, N. Uratani, and T. Ehara, "Transliteration considering context Information based on the maximum entropy method," *In Proceedings Of the IXth Machine Translation Summit*, 2003.

[10] T. Sherif and G. Kondrak, "Substring based transliteration," *In proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 944-951, 2007.

[11] J. H. Oh and K. S. Choi, "An English-Korean transliteration model using pronunciation and contextual rules," *In: Proc. Of the 19th International Conference on Computational Linguistics*, pp. 758-764, 2002.

[12] W. H. Lin and H. H. Chen, "Backward machine transliteration by learning phonetic similarity," *In: Proc. Of the Sixth Conference on Natural Language Learning*, pp. 139-145, 2002.

[13] H. Li, K. C. Sim, J.S. Kuo, and M. Dong, "Semantic transliteration of person names. *In proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 720-727, 2007.

[14] S. Bilac and H. Tanaka, "A hybrid back-transliteration system for Japanese," *In: Proc. Of the 20th International Conference on Computational Linguistics (COLING 2004)*, pp. 597-603, 2004.

[15] G. Hong, M. J. Kim, D. G. Lee, and H. C. Rim, "A Hybrid approach to English-Korean name transliteration," *In proceedings of the Named Entities Workshop, ACL-IJCNLP'09*, pp. 108-111, August 2009.

[16] N. Aswaniand and R. Gaizauskas, "A hybrid approach to align sentences and words in English-Hindi parallel corpora," *In Proceedings of the ACL Workshop on Building and Exploiting Parallel Texts*, 2005.

[17] M. G. A. Malik. "Punjabi machine transliteration," *In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pp. 1137-1144, 2006.

[18] S. ganesh, S. Harsha, P. Prasad, and V. Varma, "Statistical transliteration for cross language information retrieval using HMM alignment and CRF," *In Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, 2008, NERSSEAL Workshop, Hyderabad, India.

[19] H. Surana and A. K. Singh, "A more discerning and adaptable multilingual transliteration mechanism for Indian languages," *In Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, 2008, Hyderabad, India.

[20] T. Rama and K. Gali, "Modeling machine transliteration as a phrase based stastistical machine translation problem," *In proceedings of the Named Entities Workshop, ACL-IJCNLP'09*, pp. 124-127, August 2009.

[21] S. C. Janarthnam, S. Sethuramalingam and U. Nallasamy, "Named entity transliteration for cross-language information retrieval using compressed word format mapping algorithm," *In Proceedings of 2nd International ACM Workshop on Improving Non-English Web Searching (iNEWS08)*, CIKM-2008.

[22] A. Nayan , B. R. K. Rao, P. Singh, S. Sanyal, and R. Sanyal, "Named entity recognition for Indian languages," *In Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 97-104, 2008.

[23] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phys. Dokl.*, vol. 6, pp. 707-710, 1966.

[24] F. Gralinski, k. Jassem, and M. Marcinczuk, "An environment for named entity recognition and translation," *In Proceedings of the 13th Annual Conference of the EAMT*, pp. 89-95, 2009.

Appendix

1. Psuedo code for Leveinshtein edit distance

Input: Two strings, X and Y

Output: The minimum edit distance between X and Y

M ← length (X)

N ← length (Y)

for i = 0 to m do
dist [i, 0] ← i

for j = 0 to n do
dist [0, j] ← j

for i = 0 to m do
for j = 0 to n do

dist [i, j] = min { dist [i-1, j] + insert_cost ,
dist [i-1, j-1] + substitution_cost [X_i, Y_j],
dist [i, j-1] + deletion cost
}

End

2. Romanization scheme for mapping Roman characters and Telugu (Contains one to one configuration)

Vowels	Consonants			
a అ	k క	N ణ	l ల	౯
A ఆ	K ఖ	t త	L ణ	౯
i ఇ	g గ	T డ	v వ	
I ఊ	G ఘ	d డ	S శ	
u ఉ	f బ	D డ	R ష	
U ఊ	c చ	n న	s స	
q ఋ	C ఛ	p ప	h హ	
Q ౠ	j జ	P ఫ		
V ఎ	J య	b బ		
e ఏ	F ఙ	B భ		
E ఐ	w ట	m మ		
o ఓ	W ఠ	y య		
O ఔ	x డ	r ర		
z ఔ	X డ	Y ణ		

3. Romanization scheme for transliteration between Roman characters and Telugu (Contains one to one and many to one configuration)

Vowels	Consonants		
A అ	KA క	NA ణ	LA ల
AA ఆ	KHA ఖ	TA త	LLA ళ
I ఇ	GA గ	THA ధ	VA వ
II ఈ	GHA ఘ	DA ద	SHA శ
U ఉ	NGA జ	DHA ధ	SSA ష
UU ఊ	CA చ	NA న	SA స
R ఋ	CHA ఛ	PA ప	HA హ
RR ౠ	JA జ	PHA ఫ	
E ఎ	JHA ఝ	BA బ	
EE ఏ	NYA ఞ	BHA భ	
AI ఐ	TTA ట	MA మ	
O ఒ	TTHA ఠ	YA య	
OO ఓ	DDA డ	RA ర	
AU ఔ	DDHA ఢ	RRA రి	

Study and Improvement on Optimal Reactive Routing Protocol (ORRP) for Mobile Ad-Hoc Networks

Soma Saha¹, Tamojay Deb²

¹Women's Polytechnic, Dept. of IT, Hapania, Tripura

²Dept. of IT, Dasaratha Deb Memorial College, Khowai, Tripura
{somasaha84, tamojaydeb}@gmail.com

Abstract: In MANET, mobile nodes dynamically form temporary networks without using conventional infrastructure or centralized administration. In this paper, we have improved Optimal Reactive Routing Protocol (ORRP) [2], an existing on-demand route discovery approach that returns the shortest path in between a source-destination pair. ORRP does not use flooding. It finds the optimal route based on a cost vector. However, the protocol ORRP does not mention any effective way to compute this cost vector. This paper is a significant and two-fold extension of ORRP. We have worked on some of the basic incompleteness of ORRP. The most significant contribution is in incorporating a periodic HELLO message exchange for sensing neighborhood as well as for determination of cost vector.

Keywords: MANET, Reactive, Optimal, Secure, Beaconing, Reactive

1. Introduction

A Mobile Ad-hoc network (MANET) is a kind of wireless network where the participating nodes dynamically and arbitrarily forms a network. In such a network, each mobile node operates not only as a host but also as a router[1]. MANET is an infrastructure-less network where, there are no routers, servers, access points or cables. Participating nodes can move freely and in arbitrary ways, so it may change its location from time to time. In order to enable communication between any two nodes, a routing protocol is employed. The duty of the routing protocol in MANET is to discover the topology to ensure that each node has the recent image of the network topology to construct routes for communication. Currently, two complementary classes of routing protocols exist in the MANET world. Reactive protocols (such as AODV and DSR) acquire routes on demand, while the proactive protocols (such as OLSR, OSPF, DSDV) ensure that topological information is maintained through periodic message exchange. In both the cases it is necessary for one

mobile node to enlist atleast its neighboring nodes in forwarding a packet to its destination due to the limited transmission range of wireless network interfaces. When a node in a MANET wants to communicate with another node, which is out of the transmission range of the first node then the intermediate nodes act as router to forward the packet.

The main idea behind this paper is to study the ORRP protocol and to find out the short comings in the existing protocol. Then I have introduced a modified version of the original Optimal Reactive Routing Protocol (ORRP) [2]. In this protocol, I have added the concept of periodic HELLO message exchanging for neighbor sensing and Cost vector initialization. I am aiming to make a comparison study based on performance between the existing ORRP and the extended version proposed in this paper.

The rest of the paper is organized as follows: Section 2 presents a very brief review on reactive routing protocols for mobile ad hoc networks. In fact, as we would be working on Optimal Reactive Routing Protocol only, we need to study the existing ORRP in details. This is done in section 3. The newer version is built upon pointing the limitations and incompleteness of ORRP. Section 4 describes the proposed improved version with illustrative example to explain the operation of the improved protocol. In order to identify the improved version from the original protocol, we refer the newer one as ORRP-1 in rest of the paper. In section 5, we present a comparison study between ORRP and ORRP-1. Section 6 concludes the paper.

2. Related Works

A number of researches are done on routing protocols in MANET. I briefly outline the most relevant characteristics of them. Reactive protocols are on demand protocols that discover the route once needed (eg AODV [3]). The reactive protocols display

considerable bandwidth and overhead advantages over proactive protocols. AODV routing protocol offers quick adaptation to dynamic link conditions, low processing, low memory overheads, and low network utilization [4]. But it can't ensure loop free optimal route between two communicating nodes. Upon based on AODV several other routing protocols has been introduced. Some previous works [4] [5] [6] [7] are made on comparative study among those. Some new types of routing protocols also introduced to withstand the limitations of MANET i.e., limited bandwidth and a high degree of mobility. This work is based on ORRP [2] that loop free and optimal path every time.

3. Review on the ORRP

Optimal Reactive Routing Protocol (ORRP) is a reactive protocol that finds a loop-free, optimal path between the end nodes. In this paper I tried to expose the short comings in the existing ORRP[2] and to introduce the necessary modification. I refer the newer version of ORRP as ORRP-1 in this paper. ORRP like other reactive routing protocol is a source initiated routing algorithm. It assumes that at any given instance, any node in the network maintains a list of its neighbors and also stores the cost vectors to reach the neighboring nodes from the node. Any change in the topology including deletion of a host or a link must be communicated to the neighboring nodes.

ORRP assumes symmetric links between neighboring nodes. The working of ORRP is based on two basic operations: Procedure Update() and Procedure FindRoute(Ni,Nj). Procedure Update is responsible for neighbor sensing. If any node fails then the Update() procedure deletes the entry for that node in the list of its neighbors. Again if any node joins the network the same procedure is used for entering the proper information in appropriate place. On the other hand Procedure FindRoute(Ni,Nj) finds the shortest route between source node Ni and destination node Nj. The procedure makes use of Dijkstra's shortest path algorithm for finding path. ORRP does not compute all possible routes between a node and the remaining nodes like the proactive protocol, but it computes the shortest path from the information maintained by the participating nodes when any node wants to communicate with other node that keeps the routing overhead low.

3.1 Shortcomings of ORRP

ORRP gives innovative idea being a reactive routing protocol but need some further modification before implementation. ORRP assumes each participating nodes maintains a list of its neighbors and the corresponding cost vector for each entry. But it remains

silent regarding the value assignment for the cost vectors. ORRP does not introduce any mechanism to store the intermediate routing information. ORRP introduced two procedures. Procedure FindRoute(Ni,Nj) is responsible for finding the optimal path . Procedure Update() maintains the information regarding random topology changes. ORRP tells about cost vector in both the procedures but it does not define any procedure to assign the cost vector. Moreover ORRP needs neighbors information of each node so that the Dijkstra's algorithm can be executed but it fails to introduce any such mechanism for neighbor sensing. ORRP-1 is an effort to overcome all those incompleteness present in ORRP so that it can be implemented.

4. The Proposed ORRP-1

The basic idea behind ORRP remains unchanged in ORRP-1. But it includes few extensions to eliminate some of the deficiencies relate to the cost assignment and neighbor sensing. ORRP-1 makes use of periodic HELLO message exchange for implementing those.

4.1 Periodic beaconing in ORRP

In this approach, each node periodically broadcasts a HELLO message to its neighbors, so that each node has the local knowledge of all its neighbors and cost vector assigned to each link. Basically, neighbor sensing is the process through which a node detects changes to its neighborhood. The neighbor sensing mechanism in ORRP is designed to operate independently in the following way: each node periodically emits a HELLO-message every HELLO_INTERVAL seconds, containing the node's own address as well as the list of neighbors known to the node, including the timestamp. Upon receiving HELLO-messages, a node can thus gather information describing its neighborhood. Each node maintains an information set, describing the neighbors. Such information is considered valid for a limited period of time, and must be refreshed at least periodically to remain valid. Expired information is purged from the neighbor sets.

A node may sometimes broadcast a triggered HELLO message in response to some event that needs quick action. When a node joins an existing network it will broadcast HELLO message within its radio transmission range. The proposed protocol assumes symmetric link between neighboring nodes. The HELLO message will identify the symmetric link and will assign the cost. We propose some additional data structures towards this. The cost of link will be determined as a function of time stamp assigned by both the adjacent nodes of a symmetric link. The neighbor set of a node N_i may be defined as follows: $NS_i = \{N_j\}$, where N_j : Node

adjacent to N_i , for $I=1, \dots, m$. The following expression represents the contents of a HELLO message from N_i :

HELLO (N_i, ST_i, NS_i)

IP address of N_i	Length	ST_i
Neighbor ₁ IP address		
Neighbor ₂ IP address		

Figure 1: HELLO message

Field description

Length: The number of neighbors listed

ST_i : Time stamp of broadcasting the HELLO message by N_i .

Neighbor IP Address: The address of a neighbor. The IP addresses of the neighbors are taken from the RECORD table.

Each entry in the RECORD Table is associated with a timer. A table entry will be removed if a HELLO message from the entry's node is not received for a period of $(HELLO_LOSS)*HELLO_INTERVAL$, allowing HELLO_LOSS consecutive HELLO messages to be lost from that node. If a node don't get any HELLO message from its neighbors listed in its RECORD table for more than $(HELLO_LOSS)*HELLO_INTERVAL$ time it will discard that node from its RECORD table.

HELLO message processing

Upon receiving a HELLO message by N_j from N_i , the node N_j should update the neighbor entry corresponding to the sender node address. At first it will take the time stamp RT_j .

1. If the sender IP address does not exist in the RECORD table the receiving node it adds one entry for the sender in its RECORD table.

1.1 Then the receiver will generate another Control message to N_j and will send it back to N_i at time ST_i , where ST_j is the Time stamp of sending reply HELLO message by N_j upon receiving a HELLO message from N_i .

IP address of N_i	IP address of N_j	
ST_i	RT_j	ST_j

Figure 2: Control message

1.1.1. Node N_i will receive the control message at time RT_i . Node N_i will check whether N_j is present in its RECORD table or not. If N_j is not present it will add N_j to its record table and entry the cost associated with the link in between N_i and N_j . The cost vector C_{ij} can be

calculated using the formula:

$$C_{ij} = ((RT_j - ST_i) + (RT_i - ST_j)) / 2$$

1.1.2. If N_j is present in the RECORD table of N_i , then the cost vector will be calculated with the same formula and the entry will be updated with the newly derived C_{ij} .

2. If N_i is present in the RECORD table of N_j then also N_j will send back a reply - control message to N_i

2.1 Upon receiving the control message by N_i from N_j the same function like step 1.1 will be carried out.

Time stamp of sending Hello message	Adjacent Nodes $\{N_j\}$ for $J \in [1..m]$	Cost $C[J]$
-------------------------------------	---------------------------------------------	-------------

Figure 3: RECORD table for N_i

Depending upon the Time stamp the most recent cost vector can be identified. The protocol, as evident here, does not require maintaining any other data structure like sequence numbering etc. We can determine the most recent cost vector and update the older one with newer one from the time stamp itself.

Due to the random mobility of the network, a node may get out of the transmission range of the other nodes within the network. Then it will not receive any HELLO message. As a result the link between that node and its adjacent nodes will break down. When any node or a link fails then it will send a failure message to its entire neighbor set. Upon receiving that failure message by the adjacent nodes those nodes will delete the entry for that node from its RECORD list.

The HELLO message is sent by a node to its entire neighbor at a fixed interval of time. But when the topology of the network changes or any link between two nodes breaks then the HELLO message is generated immediately to find the current state of the network.

4.2 Procedure FindRoute(N_s, N_d)

ORRP-1 employees another procedure named **FindRoute (N_s, N_d)** [where N_s =source node and N_d =destination node] to identify the route between source and destination. EORRP ensures optimal shortest path discovery procedure as it employees Dijkstra algorithm.

For finding the routing route the source node N_s sends a message containing some special fields to its adjacent neighbor having the least weighted/cost link with the source from the RECORD table. The format of the message FindRoute(N_s, N_d) is:

1	2	3	4	5	6	7
FRP_seq	Ns	Nd	Nj	Nprev	Cprev_s	T

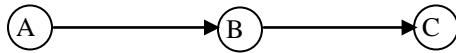
Figure 4: FindRoute() message format

FRP_seq:: The source node assigns a FRP_seq i.e. FRP sequence no which remains fixed through out the route discovery process . FRP_seq will be used during the propagation of routing route by the destination to the source.

Ns:: Ns is the source node. But the value for Ns changes every time when a HELLO message is generated by the intermediate node during the route discovery process having same FRP_seq no.

Nd:: Nd is the destination node. It remains same until reached to the original destination.

Nj: Nj is the adjacent node of the source node having the least link cost value which is the next node in the source to destination route.



Nprev: Nprev is the source node of the current source node. As for example, A is the Nprev for node C

Cprev_s: Cprev_s is the cost associate with the source node and the source node of the current source node. Here cost between A and B is Cprev_s.

Field 7:=T: The seventh field contains a token, that remains unaltered during the route finding procedure.

4.3 Algorithm for Finding Route

Step 1: The initial source node N_S assign value for FRP_seq and select the adjacent neighbor having least link cost value and sends a FindRoute() message to that node. N_S puts NULL value in field 5 & 6 of the FindRoute() message.

Step 2: Upon receiving that message by the adjacent node N_j it will check whether it is the destination node or not.
If ($N_j = N_D$) goto step 6 Else goto step 3

Step 3: If ($N_{PREV} = NULL$) then
FindRoute() message is updated.
 $N_{PREV} \leftarrow N_S$
 $N_S \leftarrow N_j$
FRP-Seq and N_D remains unaltered
 N_j is selected from the list of adjacent

nodes in RECORD table having least link cost value of newly selected source. The updated FindRoute() message is sent to the next adjacent node N_j .

Goto step 2

Else
goto step 4

Step 4: Upon receiving the message by N_j it makes a search in its RECORD table for finding the cost associate with N_S i.e. $C_{S,J}$. And also in RECORD table of N_j a search will made to find whether N_{PREV} is present in the RECORD table of N_j .

If Nprev is present in the RECORD table of N_j then

goto step 4.1

Else
goto step 4.2

Step 4.1: Consider the cost between Nprev and N_j from the RECORD table of N_j as $C_{PREV,J}$, Then perform the following:

If ($C_{PREV,J} \leq (C_{S,J} + C_{PREV,S})$) then

Store FRP_seq no and Nprev in the routing table.

Else
Store FRP_seq no and the IP address of N_s .

Step 4.2: Store FRP_seq no and the IP address of N_s .

Step 5: If ($N_j \neq N_D$) then

Update the content of the FindRoute message.

Keep values for field 1 and 3 unaltered.

$N_S \leftarrow N_j$

New N_j will be selected from the RECORD table of older N_j

If ($C_{PREV,J} \leq (C_{S,J} + C_{PREV,S})$) then

Nprev will remain unchanged.

Else

$N_{PREV} \leftarrow N_S$

$C_{PREV,S}$ will be cost between new N_S and new N_{PREV} from the RECORD table of new N_S .

Step 6: When ($N_j = N_D$) then

The destination node N_D stores the content of the seventh field i.e. T.

The acknowledgement will be backtracked to the source node from the destination node. From the cache of N_D the previous node can be obtained. The acknowledgement will be sent to the previous node.

Upon receiving ACK message by Nprev it will compare the FRP_seq no of ACK message with the FRP_ack stored in its

cache.

If both are same, that message will be delivered to its previous node stored in its cache.

Thus, the ACK message will reach the original source node and the route is discovered.

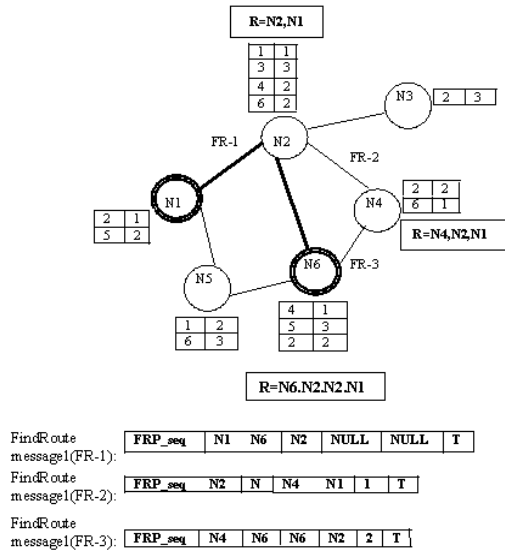


Figure 5. A configurable topology

4.3 Illustrative Example for ORRP-1

Figure 5 shows an ad hoc network with 6 host nodes. The links between each node are considered symmetric. The record table in the figure only considers two fields adjacent node and the cost determined by the HELLO message and control message exchange. N1 and N6 are source and destination respectively. N1 initiates the route finding procedures and the optimal route determined at the end of the procedure is N6, N2, N1.

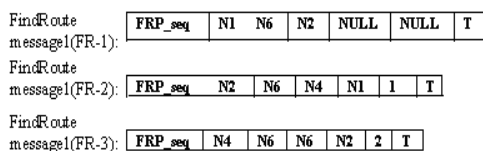
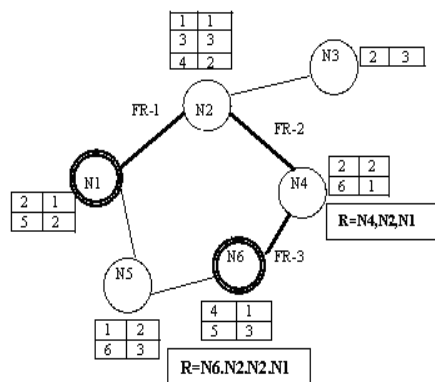


Figure 6. The revised topology

Due to mobility when node N6 comes within the transmission range of node N2 it will be sensed by N2 due to periodic HELLO message exchange and the record table will be updated accordingly. Figure 6 shows the revised topology and new route determined by FindRoute() procedure.

Likewise when any node leaves or enter the network it will be sensed by HELLO the other nodes and action will be taken accordingly.

5. Comparative Performance Study

In proactive routing protocols, the nodes keep updating their routing tables, by sending periodical messages. These tables require frequent updates for keeping the updated information regarding the network topology due to the mobility of participating nodes in MANET. A huge amount of bandwidth is wasted for periodic update of routing tables because the routing information are flooded in the whole network. These protocols require a huge amount of memory in order to maintain the global topology information by each node and require complex processing.

Reactive (On Demand) routing protocols, where routes are created only when needed. A route is established when a node wants to communicate with another one, which needs to broadcast HELLO messages that also consumes a considerable amount of bandwidth. However, even though all possible routes between each and every nodes in the network are not determined before hand the protocols are comparatively simple then the proactive one. Reactive routing protocols can determine a path every time it is executed but can't ensure the optimal path every time.

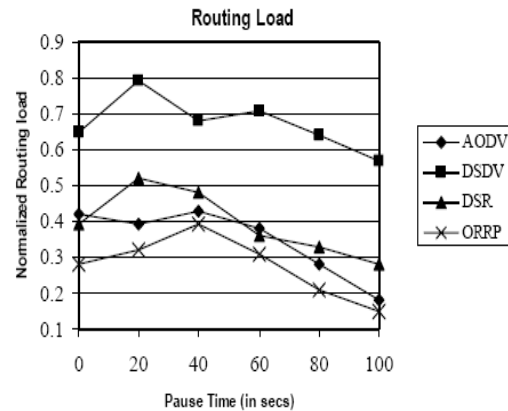


Figure 7: Study of comparative routing load

On the other hand ORRP being a reactive routing

protocol assure optimal path. ORRP uses Dijkstra's shortest path finding algorithm which always returns the shortest path within lower time bound. Only two procedures are involved in ORRP that are capable of finding the shortest path and updating of the network topology. From the implementation point of view ORRP requires some modifications that are incorporated in ORRP-1 presented in this paper.

The result of the simulation based on the routing load as the performance metric shows that the routing load for ORRP is lower than other three protocols AODV, DSR, DSDV[2]. The routing load has been measured in terms of the average number of control packets transmitted per data packet delivered at the destination. Routing load = packets sent / packets received.

ORRP-1 has all the advantages of ORRP. It also offers some more advantages. All the nodes simply keep information of its 1-hop neighbors and the cost vector of the symmetric link to each of them. To maintain this information the same HELLO message is used that are used for neighbor sensing. It involves less control message exchange. ORRP-1 does not broadcast the route request control message during route discovery, which saves a considerable amount of the bandwidth.

6. Conclusion

In this paper, the ORRP have been critically studied. Although the ORRP protocol has many advantages, major limitations have been identified towards implementing the protocol. In this paper, solutions are proposed to overcome this. We have introduced periodic HELLO message exchange. This provides an effective means to compute the cost vectors besides sensing the neighborhood. ORRP-1 is made of using the best characteristics of both reactive and proactive routing protocol. Like any other reactive routing protocols it determines the route between source and destination pair only on demand. The processing overheads in ORRP-1 are also reduced by efficient use of HELLO messaging and keeping the protocol to be executed on-demand. However it ensures the shortest path like most of the proactive routing protocols.

References

- [1] H Yang, H Y. Luo, F Ye, S W. Lu, and L Zhang, "Security in mobile ad hoc networks: Challenges and solutions" (2004). IEEE Wireless Communications. 11 (1), pp. 38-47, 2004.
- [2] N. Chaki, R. Chaki; "ORRP: Optimal Reactive Routing Protocol for Mobile Ad-Hoc Networks", Proceedings of the IEEE Int'l Conf. on Computer

- Information Systems and Industrial Management Applications (CISIM 2007), pp. 185-190, 2007.
- [3] C.E. Perkins, E. Belding Royer, and S.R. Das, "Ad hoc On demand distance vector (AODV) routing", IETF RFC 3561, July 2003.
- [4] Mohammed F. Al-Hunaity, Prince Abdullah Bin Ghazi, "A Comparative Study between Various Protocols of MANET Networks". American Journal of Applied Sciences 4 (9):pp. 663-665, 2007.
- [5] E.M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Networks," IEEE Personal Communications, vol. 6, no. 2, Apr., 1999, pp. 46-55.
- [6] Clausen, Thomas Heide ; Jacquet, P.; Viennot, L. "Comparative Study of Routing Protocols for Mobile Ad-hoc NETWORKS ".IFIP Med Hoc Net 2002
- [7] Das, S.R. Perkins, C.E. Royer, E.M, "Performance comparison of two on-demand routing protocols for adhoc networks", INFOCOM 2000. 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol.: 1, pp. 3-12, 2000

Author Biographies



Soma Saha had born at Agartala, Tripura on 2nd May, 1984.

She did his BE in Computer Sc & Engg from N.I.T., Agartala, Tripura, India in 2006 and her Masters in Computer Sc & Engg. from Tripura University, Tripura, India in 2008. Her major fields of study are in Mobile Computing, Mobile ADHOC networks.



Tamojay Deb had born at Agartala, Tripura on 12th September, 1978.

He did his BE in Computer Sc & Engg from G.B.P.E.C., H. N. B. Garhwal University, Uttarakhand, India in 2002 and his Masters in Computer Sc & Engg. from Jadavpur University, Kolkata, India in 2009. His major fields of study are in Mobile ADHOC networks and Soft-computing (Image Processing).

Simulation Environments for Wireless Sensors Networks

Basavaraj.S.M¹, V.D.Mytri² and Siddarama.R.Patil³

¹ Appa Institute of Engineering and Technology, Gulbarga, Karnataka, India

² School of Computer Science and Engineering, XYZ University,

³ P.D.A College of Engineering Gulbarga, Karnataka

Corresponding Adresses

{first author, second author, third author}@email.com

Abstract: The emergence of wireless sensor networks brought many open issues to network designers. Traditionally, the three main techniques for analyzing the performance of wired and wireless networks are analytical methods, computer simulation, and physical measurement. However, because of many constraints imposed on sensor networks, such as energy limitation, decentralized collaboration and fault tolerance, algorithms for sensor networks tend to be quite complex and usually defy analytical methods that have been proved to be fairly effective for traditional networks. Furthermore, few sensor networks have come into existence, for there are still many unsolved research problems, so measurement is virtually impossible. It appears that simulation is the only feasible approach to the quantitative analysis of sensor networks. The goal of this paper is to aid developers in the selection of an appropriate simulation tool.

1. Introduction

The goal for any simulator is to accurately model and predict the behavior of a real world environment. Developers are provided with information on feasibility and reflectivity crucial to the implementation of the system prior to investing significant time and money. This is especially true in sensor networks, where hardware may have to be purchased in large quantities and at high cost. Even with readily available sensor nodes, testing the network in the desired environment can be a time-consuming and difficult task. Simulation-based testing can help to indicate whether or not these time and monetary investments are wise. Simulation is, therefore, the most common approach to developing and testing new protocol for a sensor networks. Many published papers contain results based only on experimental simulation. There are a number of advantages to this approach: lower cost, ease of implementation, and practicality of testing large scale networks. In order to effectively develop any protocol with the help of simulation, it is important to know the different tools available and the benefits and drawbacks therein associated. Section 2 of this paper presents the problems inherent in the use of simulation for testing, specifically applied to sensor networks. Section 3 presents a number of sensor network simulators. Section 4 provides analysis, comparing the simulators in situation-specific circumstances and making recommendations to the developers of future sensor simulators.

2. Problem Formation

NS-2 perhaps the most widely used Network Simulator, has been extended to include some basic facilities to simulate sensor Networks. However, one of the problems of ns2 is its object-oriented Design that introduces much unnecessary interdependency between modules. Such interdependency sometimes makes the addition of new protocol models extremely difficult, only mastered by those who have intimate familiarity with the simulator. Being difficult to extend is not a major problem for simulators targeted at traditional networks, for there the set of popular protocols is relatively small. For example, Ethernet is widely used for wired LAN, IEEE 802.11 for wireless LAN, TCP for reliable transmission over unreliable media. For sensor networks, however, the situation is quite different. There are no such dominant protocols or algorithms and there will unlikely be any, because a sensor network is often tailored for a particular application with specific features, and it is unlikely that a single algorithm can always be the optimal one under various circumstances.

Various network simulation environments exist in which sensor networks can be tested, including GloMoSim, OPNET, EmStar, SensorSim, ns-2, and many others. However, because of the unique aspects and limitations of sensor networks, the existing network models may not lead to a complete demonstration of all that is happening [1]. In fact, the developers in charge of ns-2 provide a warning at the top of their website indicating that their system is not perfect and that their research and development is always ongoing [2]. Various problems found in different simulators include oversimplified models, lack of customization, difficulty in obtaining already existing relevant protocols, and financial cost [3]. Given the facts that simulation is not perfect and that there are a number of popular sensor simulators available, one can conclude that different simulators are appropriate and most effective in different situations. It is important for a developer to choose a simulator that fits their project, but without a working knowledge of the available simulators, this is a difficult task. Additionally, simulator developers would benefit by seeing the weaknesses of available simulators as well as the weaknesses of their own models when compared with these simulators, providing for an opportunity for improvement. For these reasons, it is beneficial to maintain a detailed description of a number of more prominent simulators available

3. Simulators

This paper will present different simulators framework. These simulators were selected based on a number of criteria including popularity, published results, and interesting characteristics and features

3.1 NS -2

NS-2 [2, 4, and 5] is the most popular simulation tool for sensor networks. It began as ns (Network Simulator) in 1989 with the purpose of general network simulation. Ns-2 is an object-oriented discrete event simulator; its modular approach has effectively made it extensible. Simulations are based on a combination of C++ and OTcl. In general, C++ is used for implementing protocols and extending the ns-2 library. OTcl is used to create and control the simulation environment itself, including the selection of output data. Simulation is run at the packet level, allowing for detailed results.

NS-2 sensor simulation is a modification of their mobile ad hoc simulation tools, with a small number of add-ons. Support is included for many of the things that make sensor networks unique, including limited hardware and power. An extension developed in 2004[4] allows for external phenomena to trigger events. Ns-2 extensibility is perhaps what has made it so popular for sensor networks. In addition to the various extensions to the simulation model, the object-oriented design of ns-2 allows for straightforward creation and use of new protocols. The combination of easy in protocol development and popularity has ensured that a high number of different protocols are publicly available, despite not be included as part of the simulator's release. Its status as the most used sensor network simulator has also encouraged further popularity, as developers would prefer to compare their work to results from the same simulator.

NS-2 does not scale well for sensor networks. This is in part due to its object-oriented design. While this is beneficial in terms of extensibility and organization, it is a hindrance on performance in environments with large numbers of nodes. Every node is its own object and can interact with every other node in the simulation, creating a large number of dependencies to be checked at every simulation interval, leading to an n^2 relationship. Another drawback to ns-2 is the lack of customization available. Packet formats, energy models, MAC protocols, and the sensing hardware models all differ from those found in most sensors. One last drawback for NS-2 is the lack of an application model. In many network environments this is not a problem, but sensor networks often contain interactions between the application level and the network protocol level.

3.2 SensorSim

SensorSim is a simulation framework for modeling sensor networks. It is build upon on the NS-2 simulator and provides additional features for modeling sensor networks . SensorSim [6] uses ns-2 as a base, and extends it in three important ways. First, it includes an advanced power model. The model takes into account each of the hardware components that would need battery power in order to

operate. The developers researched the affects of each of these different components on energy consumption in order to create their power model. It is included as part of the sensor node model (Figure 1).

Secondly, SensorSim includes a sensor channel. This was a precursor to the phenomena introduced to ns-2 in 2004. Both function in approximately the same way. SensorSim's model is slightly more complicated and includes sensing through both a geophone and a microphone. However, the model is still simplistic, and the developers felt that another means of including more realistic events was needed.

This led to the third extension to ns-2: an interaction mechanism with external applications. The main purpose is to interact with actual sensor node networks. This allows for real sensed events to trigger reactions within the simulated environment. In order to accomplish this, each real node is given a stack in the simulation environment. The real node is then connected to the simulator via a proxy, which provides the necessary mechanism for interaction.

One further extension to ns-2 is the use of a piece of middleware called Sensor Ware. This middleware makes it possible to dynamically manage nodes in simulation. This provides the user with the ability to provide the network with small application scripts than can be dynamically moved throughout the network. This ensures that it is not necessary to preinstall all possible applications needed by each node, and provides a mechanism for distributed computation. Because of the battery model and sensor channel, improvements were made in the associated hardware models when compared with ns-2. However, especially in the case of the sensing hardware, the models are still very simple and do not accurately reflect what is found on most sensors. Like ns-2, SensorSim faces a scalability problem. Additionally, SensorSim is not being maintained and is not currently available to the public.

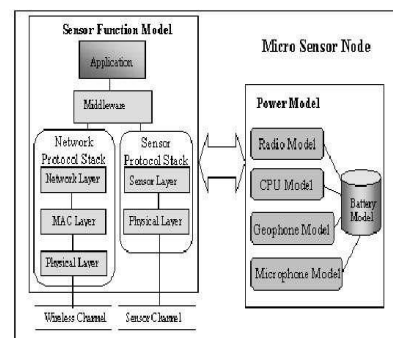


Figure 1. Micro sensor node model in SensorSim

3.3 SENSE

The SENSE is designed to be an efficient and powerful sensor network simulator that is also easy of use. The SENSE [7] simulator is influenced by three other models. It attempts to implement the same functionality as ns-2. However, it breaks away from the object-oriented approach, using component based architecture. It also includes support for parallelization. Through its component-based model and support for parallelization, the developers attempt to address what they consider to be the three most critical factors in simulation: extensibility, reusability, and scalability. SENSE was developed in C++, on top of COST, a general purpose

discrete event simulator. It implements sensors as a collection of static components. Connections between each component are in the format of in ports and out ports (Figure 2). This allows for independence between components and enables straightforward extensibility and reusability. Traversing the ports are packets. Each packet is composed of different layers for each layer in the sensor. The designers try to improve scalability by having all sensors use the same packet in memory, assuming that the packet should not have to be modified. SENSE's packet sharing model is an improvement on ns-2 and other object-oriented models that can not do this, helping improve scalability by reducing memory use. However, the model is simplistic and places some communication limitations on the user. While SENSE implements the same basic functionality as ns-2, it can not match the extensions to the ns-2 model. Whether because it is a new simulator, or because it has not achieved the popularity of ns-2, there has not been research into adding a sensing model, eliminating physical phenomena and environmental effects.

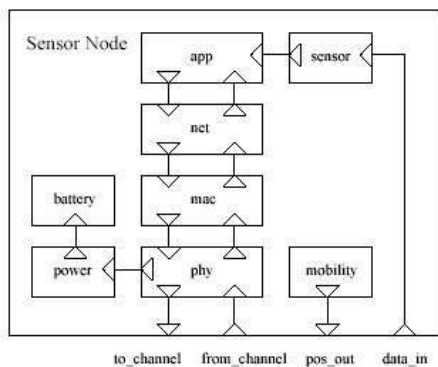


Figure 2. SENSE's sensor node structure with ports

3.4 Mannasim

Mannasim[9] goal is to develop a detailed simulation framework, which can accurately model different sensor nodes and applications while providing a versatile test bed for algorithms and protocols. Numerous challenges make the study of real deployed sensor networks very difficult and financially infeasible. At the current stage of the technology, a practical way to study WSNs is through simulations that can provide a meaningful perspective of the behavior and performance of various algorithm. This framework is free software and it can be redistributed under the GNU Public License

Mannasim is a Wireless Sensor Networks simulation environment comprised of two solutions: The Mannasim Framework, The Script Generator Tool

The Mannasim Framework is a module for WSN simulation based on the Network Simulator (NS-2). Mannasim extends NS-2 introducing new modules for design, development and analysis, development and analysis of different WSN applications. The Script Generator Tool (SGT) is a front-end for TCL simulation scripts easy creation. SGT comes blunded with Mannasim Framework and it's written in pure Java making it platform independent

3.5 EYES WSN Simulation Framework

At the start of the EYES WSN[10] project the template was needed to be built because the OMNeT++ simulator did not include support for mobile networks that communicate using radios. Although the existing ones are quite complicated to use, we tried to build a simple simulation framework and we have recently extended it with a language translator tool named NesCT. With the benefit of this tool we are able to run most of the code written in TinyOS[11] using Omnet++[12] and our simulation framework. This tool is a general purpose language translator and with some trivial customization it's also possible to make it work with other environments too. The framework was designed in such a way that allows easy modifications of the main parameters and, at the same time, the implementation details are transparent to the user.

Mobility is implemented (Random Way Point algorithm by default). Each node is responsible for defining its own trajectory and announcing it to the simulator. Nodes exchange messages using wireless communication. A message will be heard by all the neighbours situated within the transmission range (the modules within transmission range are connected automatically to each-other). The user can specify if unidirectional or bidirectional links have to be used. Each node can specify and update its transmission range independently. The nodes have different kinds of failing probabilities (individual failures, failures that affect regions of the map, etc.) Maps for area failures can be specified and used. Other maps can easily used for obstacles, fading, etc. In order to perform all of this features we have chosen to use.

3.6 NS-2 MIUN

NS-2 is a popular Open source Network Simulator. A lot of researchers in the community of Wireless Sensor Networks have used ns2 to verify their research results. However, ns2 is not an Easy tool for the simulation of Sensor Networks, partially because of its high difficulty in understanding the ns2 itself, and also because there is currently lack of support for sensor network simulation.

Ns modified to support wireless sensor network simulation, with a specialty on intrusion detection simulation. This enhanced parts is named NS2-MIUN[13] The enhanced features include.

The Integration of **NRL's phenomenon node**, which enables the ability of simulating an environmental phenomenon.

The Integration of **AODVUU**, which is an AODV Routing Protocol implementation that follows AODV specification better than the one included in the Standard ns2 Release.

The definition of a new packet type PT_SensorApp, which is used to simulate the type of packets used by sensor application.

The support of dynamic packet destination configuration. In the standard ns2 release, the <src, dst> pair is configured by statically binding an agent in the source node with an agent in the destination node in the TCL scenario file. This means a source node needs to configure multiple source agents when there are multiple potential recipients and bind

each potential <src, dst> pair manually at the configuration file. This doesn't scale well in a dynamic wireless sensor network, where the destination node can vary over time. This drawback is fixed by allowing run-time <src, dst> binding.

The integration of an intrusion detection module. It is a module inserted between the MAC layer and the network layer that captures all packets and impose intrusion detection analysis. The imitation of different attacks. The attacks implemented include wormhole, symbol / ID spoofing, DOS / DDOS, sinkhole, etc. An Extension for Simulating multi-homed nodes [14]ns-2 also is provided.

4. Analysis

This paper by no means presents an exhaustive list of sensor simulators. But the most of the issues facing the developers of sensor networks can be seen in this paper. Of course, many decisions must be made for specific situations rather than following all encompassing guidelines.

The developers must decide whether they want a simulator or an emulator. Each has advantages and disadvantages, and each is appropriate in different situations. Generally, a simulator is more useful when looking at things from a high view. The effect of routing protocols, topology, and data aggregation can be see best at a top level and would be more appropriate for simulation. Emulation is more useful for fine-tuning and looking at low-level results. Emulators are effective for timing interactions between nodes and for fine tuning network level and sensor algorithms.

If the developers decide to build a simulator, another design level decision that must be made is whether to build their simulator on top of an existing general simulator or to create their own model. If development time is limited or there is one very specific feature that the developers would like to use that is not available, then it may be best to build on top of an existing simulator. However, if there is available development time and the developers feel that they have a design that would be more effective in terms of scalability, execution speed, features, or another idea, then building a simulator from the base to the top would be most effective.

In building a simulator from the bottom up, many choices need to be made. Developers must consider the pros and cons of different programming languages, the means in which simulation is driven (event vs. time based), component-based or object oriented architecture, the level of complexity of the simulator, features to include and not include, use of parallel execution, ability to interact with real nodes, and other design choices. While design language choices are outside of the scope of this paper, there are some guidelines that appear upon looking at a number of already existing simulators.

Most simulators use a discrete event engine for efficiency. Component-based architectures scale significantly better than object-oriented architectures, but may be more difficult to implement in a modularized way. Defining each sensor as its own object ensures independence amongst the nodes. The ease of swapping in new algorithms for different protocols also appears to be easier in object-oriented designs. However, with careful programming, component based architectures perform better and are more effective. Generally, the level of complexity built into the simulator

has a lot to do with the goals of the developers and the time constraints imposed. Using a simple MAC protocol may suffice in most instances, and only providing one saves significant amounts of time. Other design choices are dependent on intended situation, programmer ability, and available design time.

5. Conclusion

The goals of this paper were to provide background on a number of different sensor network simulators and present the best and worst features of each. The purpose was three-fold. First, knowing the strengths and weaknesses of a number of different simulators is valuable because it allows users to select the one most appropriate for their testing. Second, the developers of new simulators are well served knowing what has worked in previous simulators and what has not. It also allows user to know how to scale NS-2 to suite their problem for simulating sensor networks

References

- [1] S. Park, A. Savvides and M. B. Srivastava. Simulating Networks of Wireless Sensors. *Proceedings of the 2001 Winter Simulation Conference*, 2001.
- [2] The Network Simulator – ns-2. <http://www.isi.edu/nsnam/ns>.
- [3] Vlado Handziski, Andreas Köpke, Holger Karl, and Adam Wolisz. "A Common Wireless Sensor Network Architecture?". *Senzornetze*, July 2003.
- [4] Ian Downard. Simulating Sensor Networks in ns-2. *NRL Formal Report 5522*, April, 2004.
- [5] Valeri Naoumov and Thomas Gross. Simulation of Large Ad Hoc Networks. *ACM MSWiM*, 2003.
- [6] Sung Park, Andreas Savvides, and Mani B. Srivastava. SensorSim: A Simulation Framework for Sensor Networks. *ACM MSWiM*, August, 2000.
- [7] Gilbert Chen, Joel Branch, Michael Pflug, Lijuan Zhu, and Boleslaw Szymanski. SENSE: A Sensor Network Simulator. *Advances in Pervasive Computing and Networking*, 2004.
- [8] Jonathan Pollet, et al. ATEMU: A Fine-Grained Sensor Network Simulator. *Proceedings of SECON'04, First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2004.
- [9] The Mannasim Framework <http://www.mannasim.dcc.ufmg.br/>
- [10] Eye WSN Framework <http://wwwes.cs.utwente.nl/ewnsim/>
- [11] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. *Proceedings of SenSys'03, First ACM Conference on Embedded Networked Sensor Systems*, 2003.
- [12] C. Mallanda, A. Suri, V. Kunchakarra, S.S. Iyengar, R. Kannan, and A. Durresi. Simulating Wireless Sensor Networks with OMNeT++.
- [13] NI-MIUM <http://apachepersonal.miun.se/~qinwan/resources.htm>.
- [14] Simulating Wireless Multihomed Node in NS-2. Qinghua Wang, Tingting Zhang, Department of Information Technology and Media, Mid Sweden University, Sundsvall, SE 85170, Sweden

A Video Sharing Platform for mobile devices using Data Grid Technology.

Sandip Tukaram Shingade¹, Pramila M Chawan²

Computer Engg Department,

VJTI, Mumbai

shingadesandip@gmail.com¹

pmchawan@vjti.org.in²

Abstract:

In wireless network there is limitation of storages space and characteristics their will extraordinary challenges to sharing the video files for mobile devices. To solve this problem, we use Mobile grid system for wireless network and P2P protocol and also propose architecture to establish video file sharing platform for mobile devices. We sharing video file from mobile devices using Index server to Node server for client mobile device.

Keywords: *Grids, peer-to-peer systems, Replica Location Service, resource discovery service Stream Data Processing.*

1. Introduction:

Sharing creates new possibility of entire world and human life, the sharing that we are concerned with is primarily file exchange and also direct access to computers, software, data, and other resources, as is required by a range of collaborative problem solving and resource-brokering strategies emerging in industry, science, and engineering. File sharing is necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. Wireless grids Better use or resources improved energy, power and spectral efficiency. It is very difficult to download the file in our mobile due to the wireless network limit, unstable characteristic and restricted storage space, and so mobile users face challenges in establishing connections with other users for sharing video, files. Internet users need a place to share their video clips.

YouTube saw their demand and becomes the pioneer of video sharing website. Users can establish personal theaters, movie issue stations, and news stations in YouTube to substitution tradition communication media

The remaining of this paper is organized as follows: section 2 explains the background of file sharing mobile. Section 3 Explains Different Data Transfer techniques for mobile devices .Section 4 Explain Different modules used by system. Section 5 Explain Administration System. Section 6 Explains UML Diagrams for the system .Section 7 Explains Language used by System implementation. Section 8 Explains Implementation snapshot for the System .Section 9 Explains Conclusion and Future section .10 gives the References.

2. Background:

For developing P2P collaborative application in a mobile devices ad hoc networking devices, and close mobile devices establish a cooperative while they are also connected to the cellular network. One of the most critical characteristics of the mobile grid system is the intermittent connectivity of mobile devices. We can find similar situations in Peer-to-Peer computing area. In general, P2P system consists of huge number of computing devices and they can act either as a client or a server. In P2P, each machine's CPU cycles, storages, and contents can be shared in order to broaden their resource limitations

3. Different data transfer techniques for mobile devices:

3.1 Clint to Server: Well known, powerful, reliable server is a data source. Clients request data from server. Very successful model for WWW (HTTP), FTP and Web services. But the limitation of client and server is Scalability is hard to achieve, Presents a single point of failure , Requires administration , Unused resources at the network edge

3.2 Peer-to-Peer Protocol : Peer to Peer networks is that all clients provides bandwidth, storage space and computing power .Simply it means network of peer node acting as both server and clients For mobile devices it include: a)Short connection time b)Decreased levels of user interaction

3.3 Data Grid: Data Grids are grid computing systems that deal with data. They are built on next-generation computing infrastructures, providing intensive computation and analysis of shared large-scale databases, from hundreds of terabytes to petabytes, across widely distributed scientific communities. We adopted the Globus Toolkit as our Data Grid Infrastructure. The Data Grid environment provides solutions for security, video and data management, as well as information services .

3.4 The Globus project : To building grid system and application there is use Globus toolkit is an open source software toolkit. The Globus Toolkit developed within the Globus project provides middleware services for Grid computing environments. Major components include the Grid Security, Infrastructure (GSI), which provides public-key-based authentication and authorization services; resource management services, which provide a language for specifying application requirements, mechanisms for immediate and advance reservations of Grid resources, and for remote job management; and information services.

3.5 .net: It is used in internal domain name system.

3.6 Java CoG Kit: It combines Java technology with Grid Computing to develop advanced grid services and basic Globus resource accessibility.

4. Different modules used by system:

4.1. Client Module: This module will be implemented in J2ME used to connect to the Index Server running in web server (Tomcat). The client (J2ME) will be processed by the user by his user menu whether to upload or to download a file.The user can upload or to download a Text, Image file from the server by sharing the resources directly to another client through the server, in order to reduce wireless network limit, unstable characteristic and restricted storage space, so mobile users face challenges in establishing connections with other users for sharing video, image, text files.

4.2. Index Server Module: Index Server responsibility is to calculate the Work Load of the Server nodes (where the files are stored) and it will calculate the which server node is very effective by its least working load, so that the client's request can be forwarded to that Server node. In our scheme, after users log into the index server through hard-wired or wireless networks, the index server based on the loading on each server node will assign them to grid server nodes. Users can look up the file databases to find out videos they want, and download the file from the server.

4.3. Server Nodes Module: Server node process the redirected request from indexed server and sends the response to the client directly. Using GPRS connection.

5. File sharing Administration System:

5.1 Resource Sharing:

Resource sharing it gives resource requesters login to the index server through hard wired or wireless

network. User can see resource list database to find resources they want and where to connect to the user who owns the resource, and what other users also downloaded the resource and what other users also downloaded the resource from the server.

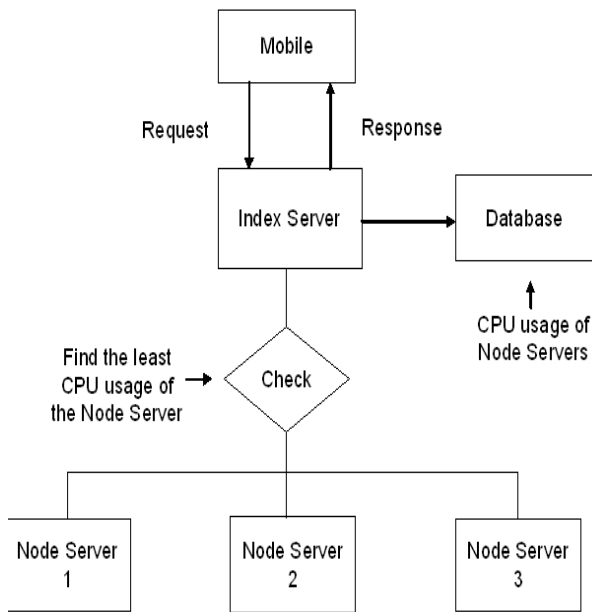


Figure 1: Administration System

5.2 Management: It consists of three part (1) Information Monitor, (2) Replica Manager, and (3) Data Transfer manager,

Information Monitor: Administrators can monitor the operational status of each machine through the System Information. Monitor being integrated into the Interface Manager. When unusual events occur, the System Information Monitor notifies the Administrator to respond appropriately, thus improving service satisfaction and productivity.

Replica Management: It can create and delete replicas at specified storage sites. A replica manager typically maintains a replica catalog containing replica site addresses and file instances. The Replica Manager periodically synchronizes data lists on all grid servers to ensure data list identical. If the access frequency of some files is high, the Replica Manager will save the files on grid servers, and

delete them when access frequency is lower than a given threshold.

Data Transfer Management: It is responsible for data in data-intensive applications. it provides effective and secure transmission for users.

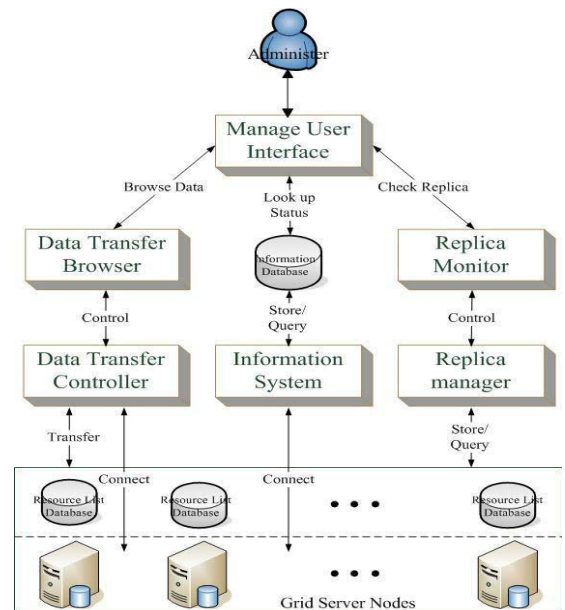


Figure 2: Administration Operation

5.3 Video Download and Upload:

In our scheme, after users log into the index server through hard-wired or wireless networks, they will be assigned to grid server nodes by the index server based on the loading on each server node. Users can look up the video databases to find out videos they want, and download the video from the server. We also provide the sharing method, if users want to share videos to other users. Users only need to upload their video to our servers; the Video Format Converter will convert them to Flash format and enroll them in the video databases. All videos are transmitted using the GridFTP protocol. The video download and upload scenario is depicted in Figure 3.

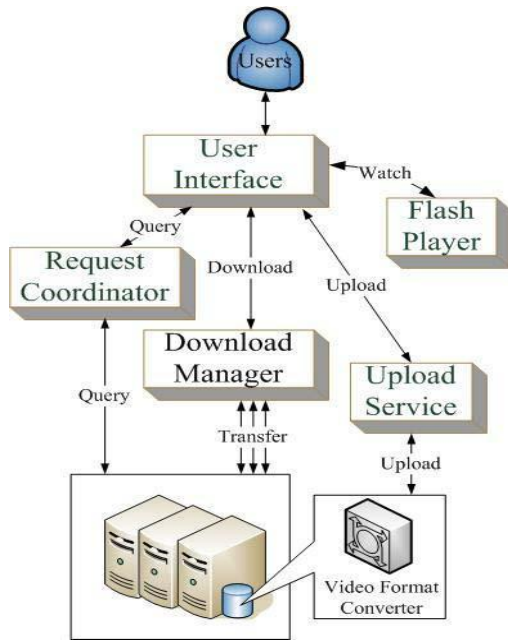


Figure 3: Operation of Client Devices.

5.4 Download Strategy

Our system is a video sharing system as well as Gnutella, Napster, and many peer-to-peer network systems. Users can download file which they want and upload the possessive file which other users need at the same time. Our destination is fast network sharing for let anyone can fast get any files. To attain to multi-point download and resume broken download, files are divided in full chunks of 9,728,000 bytes plus a remainder chunk. Furthermore, valid downloaded chunks are available for sharing before the rest of the file is downloaded, speeding up the distribution of large files throughout the network. The system is designed for users to search the videos for users need. We also designed a management interface, with an integrated data transfer service, replica manager, and information monitor to facilitate user operation of the system. When uploads each chunks, sharer are gave a time T. If a chunk has not shared completely in the time, it will be gave to other people have this chunks to share. Avoid some sharer's network speed is too slow to cause the

entire download speed to reduce (see Figure 4).

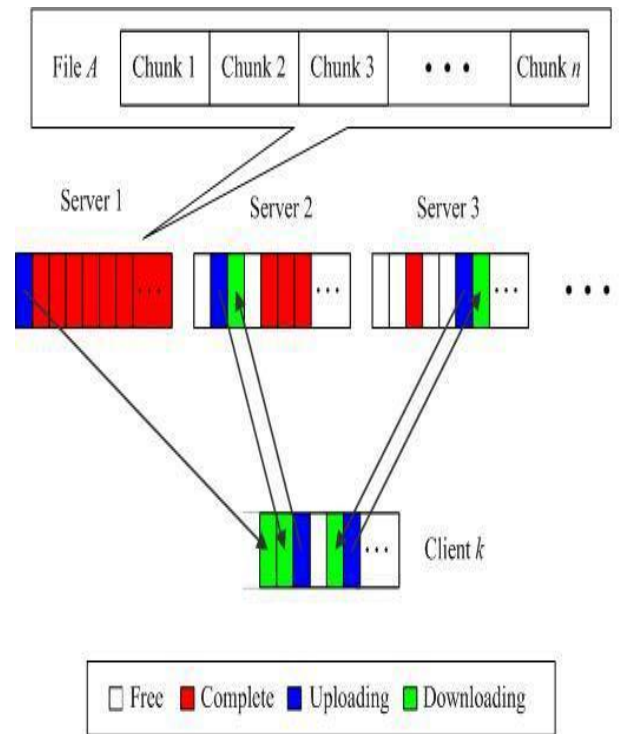


Figure 4: Video Download strategy

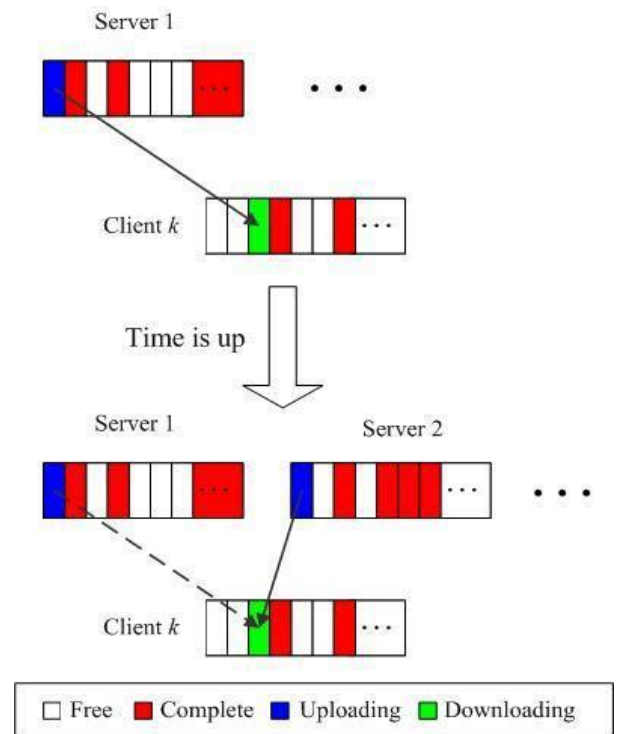


Figure 5: Time of upload chunk is up

5. UML Diagrams for the system.

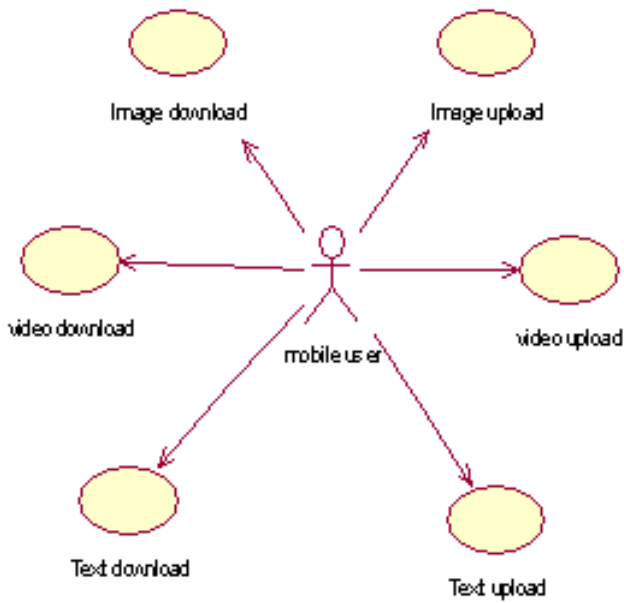


Figure 6: Use case Diagram for the mobile user system

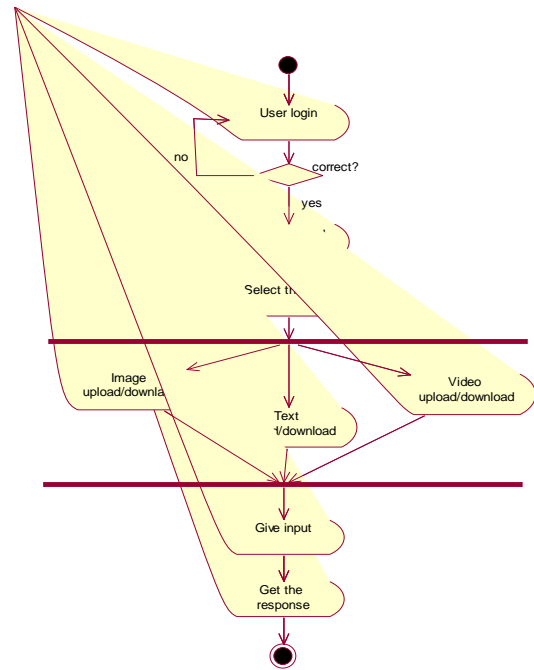


Figure 8: Activity Diagram for the system

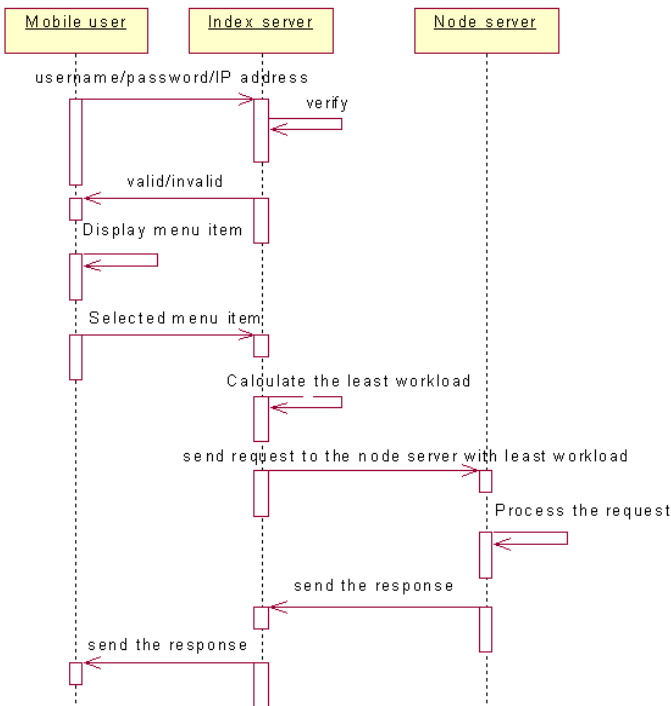


Figure 7: Sequence Diagram for the system

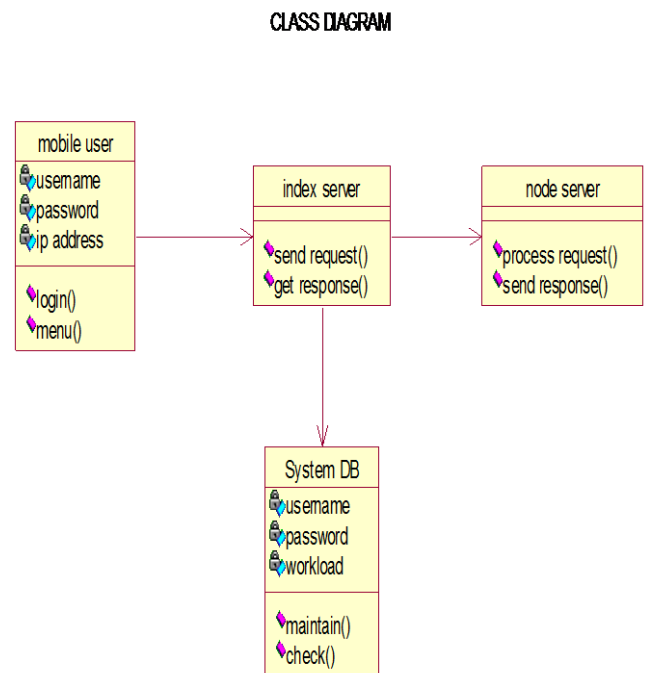


Figure 9: Class Diagram for the system

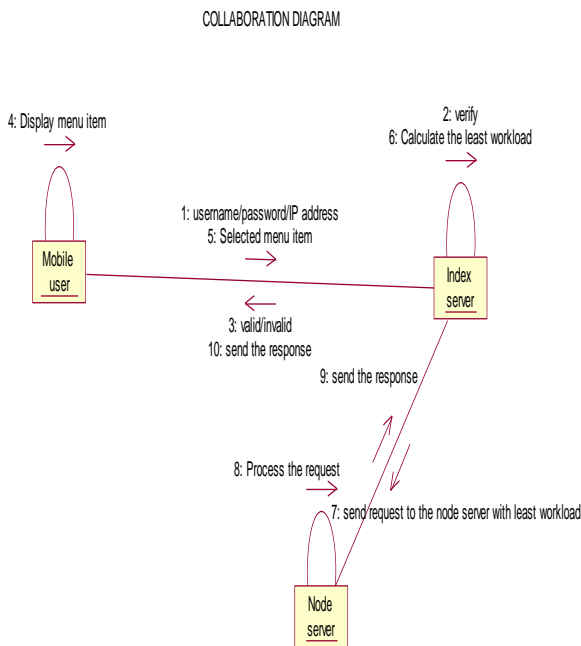


Figure 10: Collaboration Diagram for the system

7. Language used by System.

7.1 J2ME: The Java™ 2 Platform, Micro Edition (J2ME™) Wireless Toolkit is a state-of-the-art tool for developing wireless applications using the Java programming language. The toolkit is an emulation environment for developing applications targeted at J2ME Connected Limited Device Configuration (CLDC)/Mobile Information Device Profile (MIDP) technology-enabled mobile devices. Developers using the J2ME Wireless Toolkit can rest assured that their applications are compatible with CLDC/MIDP J2ME implementations. And with the toolkit’s performance optimization and tuning features, they can quickly bring to market efficient and successful wireless applications.

7.2 J2EE: J2EE stands for Java 2 Enterprise Edition for applications which run on servers, for example, websites.

7.3 JMF: The Java Media Framework API (JMF) enables audio, video, and other time-based media to

be added to applications and applets built on Java Technology. This optional package, which can capture, playback, stream, and transcode multiple media formats, extends the Java 2 Platform Standard Edition (J2SE) for multimedia developers by providing a powerful toolkit to develop scalable, cross – platform technology.

8. Implementation snapshot for the System.

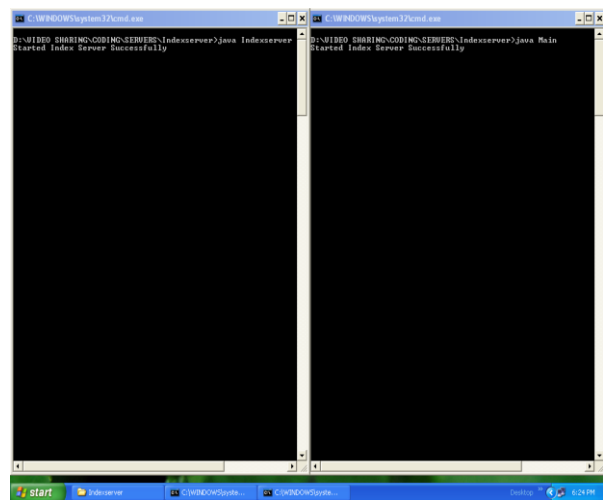


Figure 11: Initializing Index Server

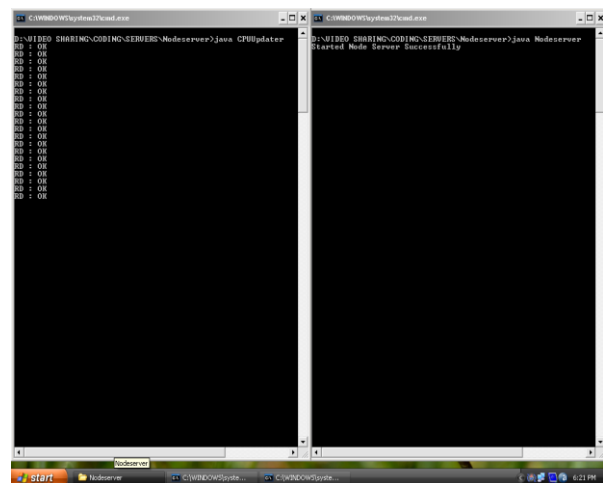


Figure 12: Initializing Node Server

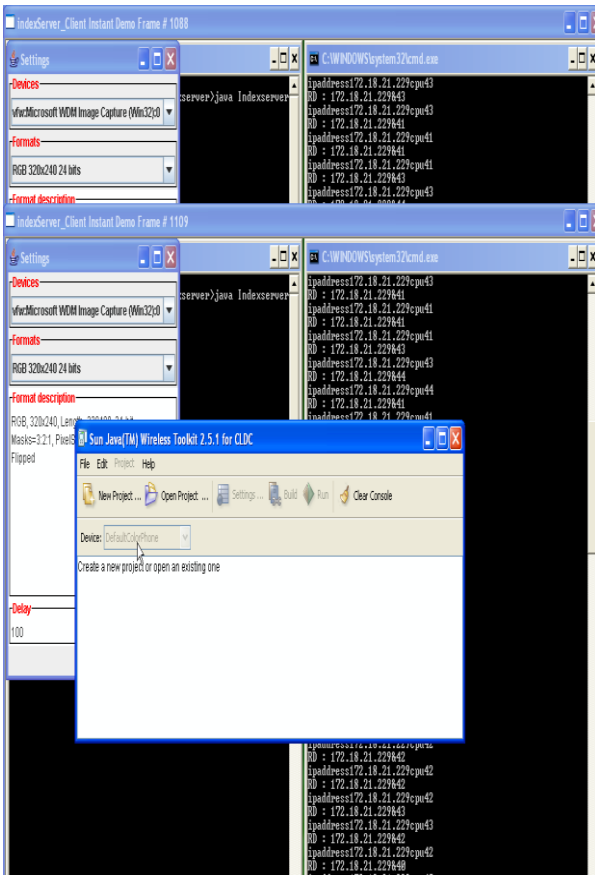


Figure 17: Open “Grid Video Project”

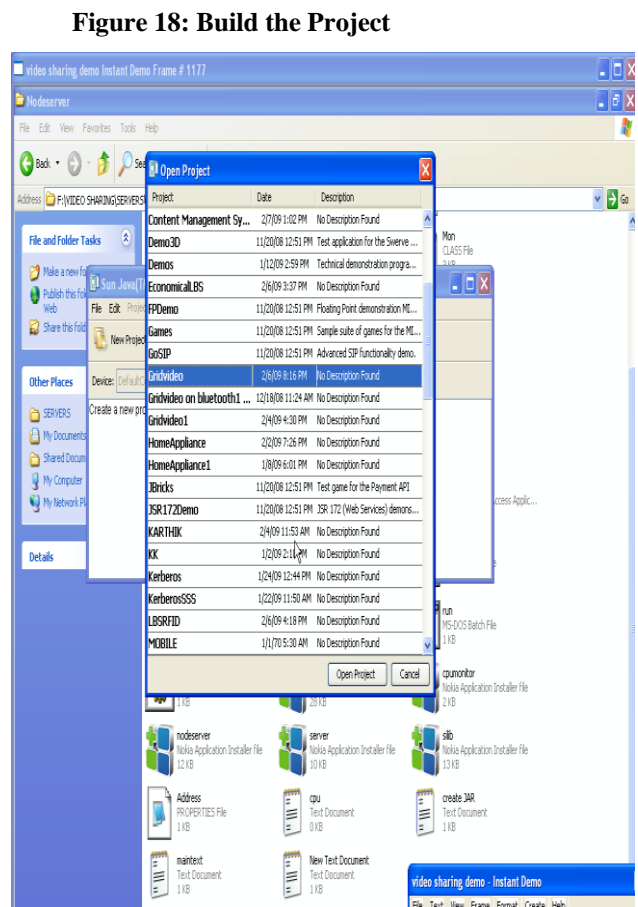


Figure 19: Build the Grid Video Project

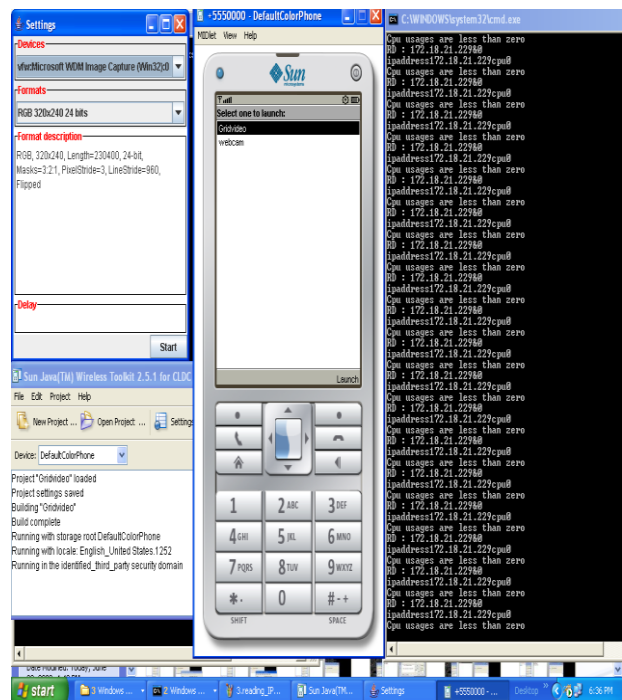
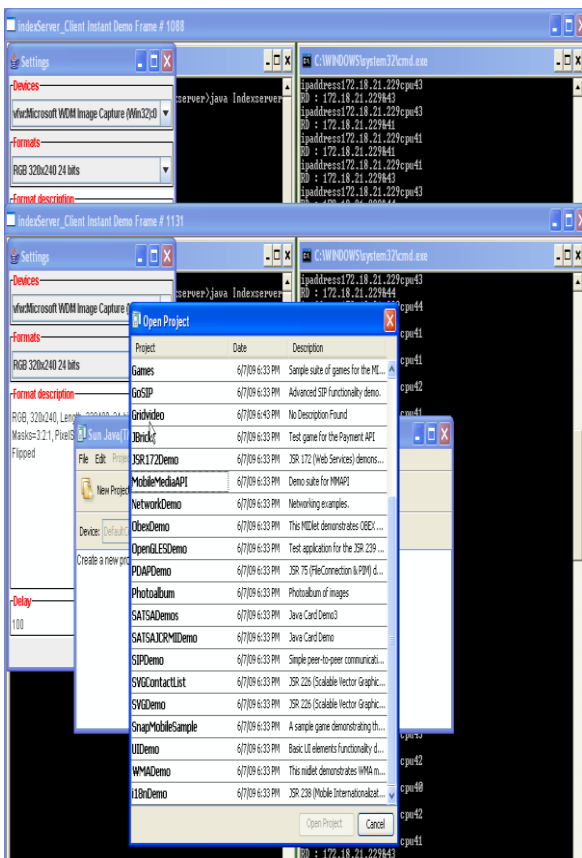


Figure 20: Using Web Camera



Figure 21: Video Mode

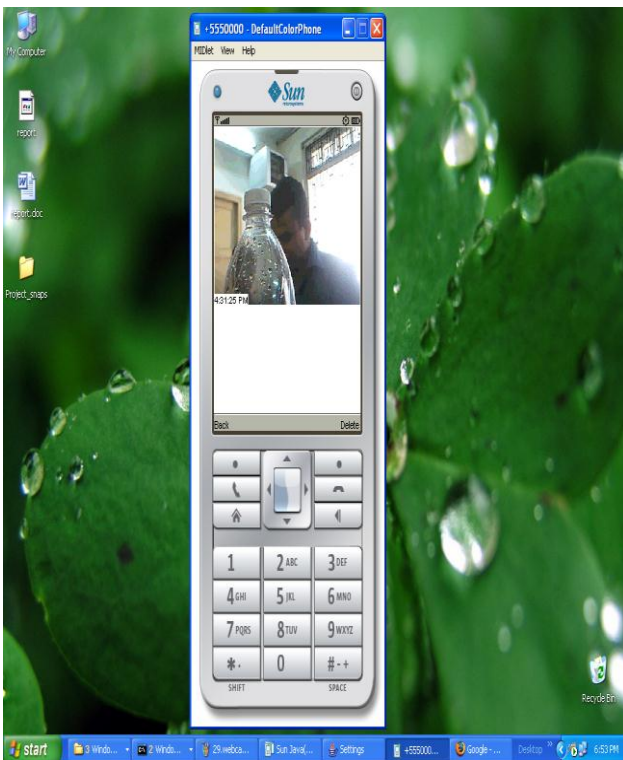


Figure 22: Playing Video

9. Conclusion and Future:

This paper designed video file sharing architecture and describe file sharing Administration System. In this paper we have presented Video File requester can collect the limited sharing traffic to increase download speed, and replaces immediately supplies the files origin. We use Data Grid system to handle some central problems, i.e., search file location and client certification. Data Grid is extendable, let us can easily connect with many store equipment to form a large-scale storage system. Our system also depend on its' computing power to classify, analyze, and convert various kinds of video files, to keep the files in the newest state at all times. This technology we have seen that using data grid can be used for carrying video streaming simulation.

10. Reference:

- [1] D Janakriram, "Grid Computing"
- [2] Globus Project, <http://www.globus.org/>
- [3] XML, <http://www.w3.org/XML/>
- [4] BioComfort Health Manager. Available: <http://www.biocomfort.com>
- [5] P2P Networks, <http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/Intro.html>
- [6] Medintegra. Available: http://www.telemedicineindia.com/medint_web.html
- [7] B. Segal, "Grid Computing, The European Data Project," IEEE Nuclear Science Symposium and Medical Imaging Conference, Lyon, 15-20 October 2000, pp.2/1.
- [8] R.S. Chang and J.S. Chang, "Adaptable Replica Consistency Service for Data Grids," *Third International Conference on Information Technology: new Generations (ITNG '06)*, pp. 646-651, 2006.
- [9] FFmpeg, <http://ffmpeg.mplayerhq.hu/>
- [10] FLVTool2, <http://rubyforge.org/projects/flvtool2/>

Author Biographies



Shingade Sandip Tukaram is currently doing her M.Tech at “Veermata Jijabai Technological Institute ,Matunga , Mumbai (INDIA) and received Bachelors’ Degree in Computer

Engineering from “Vishwakarm Institute of Technology “ Pune (INDIA) in 2007. His areas of interest are Software Engineering and Database management System. He has authored Two National and Two International papers in Conferences.



Pramila M.Chawan is currently working as an Assistant Professor in the Computer Technology Department of “Veermata Jijabai Technological Institute (V. J. T. I.), Matunga, Mumbai (INDIA)”. She received her Masters’ Degree

in Computer Engineering from V. J. T. I., Mumbai University (INDIA) in 1997 & Bachelors’ Degree in Computer Engineering from V. J. T. I., Mumbai University (INDIA) in 1991 .She has an academic experience of 18 years (since 1992). She has taught Computer related subjects at both the (undergraduate & post graduate)

Predictive Preemptive Ad Hoc on-Demand Multipath Distance Vector Routing Protocol

Sujata.Mallapur¹

¹Appa Institute of Engineering and Technology, Gulbarga.
Sujatank000@yahoo.co.in

Abstract: Routing in Ad-Hoc network is a challenging problem because nodes are mobile and links are continuously created and broken. Routing protocol for Ad-Hoc networks uses the route discovery and route maintenance mechanisms, the performance of these routing protocol depends on these mechanisms. In the existing on-demand protocols the route discovery algorithm initiates the route discovery after the path breaks, so it creates the frequent route discovery and route failure problem. To avoid this problem, in this paper we propose a Predictive Preemptive approach to route discovery. Route discovery is initiated when a “route failure” is about to occur rather than waiting for the break to happen. Proposed predictive preemptive routing protocol predicts the route failure by the received power of the packet. To evaluate this approach added it to the AOMDV routing protocol, and evaluated its performance with AOMDV. The Predictive Preemptive AOMDV was implemented using NS-2. The simulation results show the Proposed approach improve the performance. It reduces the routing overhead, decreases the route discovery, delay and improves PDF.

Keywords: Ad-Hoc networks, AODV, AOMDV, Multipath Routing, PPAOMDV.

1. Introduction

A mobile Ad-Hoc Network (MANET) [1] is a collection of mobile nodes that form a wireless network without the use of any fixed base station. Each node acting as both a host and a router arbitrarily and communicates with others via multiple wireless links, therefore the network topology changes greatly. The routing protocols proposed so far can be divided in to 2 categories: proactive routing protocol and reactive routing protocol. Reactive routing protocol, which initiates route computation only on demand, performs better than proactive routing protocol, which always maintains route to destination by periodically updating, due to its control overhead.

In such dynamic network, it is an essential to get route in time, perform the routing with maximal throughput and minimal control overhead. Several on-Demand routing protocol have been proposed. In such protocols, nodes build and maintain the routes as they are needed. Examples of these protocols include the Dynamic Source Routing (DSR) [8] and Ad hoc On-Demand Distance Vector AODV Routing protocol [2]. These protocols initiate a route discovery process whenever a node a route to a particular destination. In AODV the source broadcasts a route Request (RREQ) packet in the network to search for route to the destination. When a RREQ reaches either the destination or an intermediate node that knows a route to the destination, a Route Reply (RREP) packet is unicast back to the source.

This establishes a path between the source and destination. Data is transferred along this path until one of the links in the path breaks due to node mobility. The source is informed of this link failure by means of a Route Error (RERR) packet from the node upstream of the failed link. The source node then re-initiates a route discovery process to find a new route to the destination. It is a single path routing, which needs new route discovery whenever path breaks. To overcome such inefficiency, several studies have been presented to compute multiple paths. If primary path breaks, they provide alternative paths to send data packets without executing a new discovery.

The current multipath routing protocols multiple route obtained during Route Discovery process [12]. The best path that is the path with the shortest hop count is chosen as the primary path for data transfer while other paths are used only when primary path fails. These protocols do not perform any prediction of route failure before the path breaks or route fails. As a result it leads the problem of frequent route discovery for data transmission.

In this paper, we propose an approach that uses the “Route Failure Prediction Technique” for estimating whether an active link is about to fail or will fail. To evaluate this approach to route failure prediction, we have added it to Ad Hoc on- Demand Multipath Distance Vector Routing Protocol (AOMDV) using the Network simulator. This simulator includes implementations for many ad hoc routing protocols, and it has been validated by the frequent use by researchers.

In the existing On-Demand Routing Protocol the Route Discovery is initiated when all the path breaks, waiting to break the path leads to problem of frequent route discovery and increases the delay. To solve this problem it computes “the received power of each receiver node” using route failure prediction technique to discover the route from source and destination. A source node sends the RREQ packet, if the receiver node is less than threshold value then drops the packet and sends the warning packet to the sender than a source node discards the route containing, therefore, the selected routing path exclude the nodes that are going out of threshold. Similarly, to send the DATA packet it again calculates the received power of the receiver node if it is less than threshold sends RERR packet. A Route Maintenance mechanism is implemented to repair a broken route or finds the new route when all route failed. A Route Maintenance mechanism is implemented to repair a broken route or finds the new route when all route failed.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 describes the AOMDV

route discovery and route maintenance. The proposed protocol is presented in section 4, and its performance is evaluated and compared with that of AOMDV in section 5. Some conclusions are given in section 6.

2. Related Work

This section summarizes various examples of on-demand multipath routing protocols especially from the viewpoint of route discovery strategy

AODV Backup Routing (AODV-BR) [3] enhances the AODV by letting each neighboring node of a primary route maintain its own backup route. When the node over a primary route detects a link failure, it transmits a route update message and a neighboring node receiving this message activates the backup route. A problem of this approach is limitation of the route selection that is at most within one hop distance.

Preemptive Ad Hoc routing Protocol [12] propose a preemptive route maintenance extension to on-demand routing protocols. Its aim is to find an alternative path before the cost of a link failure is incurred. The received transmission power is used to estimate when a link is expected to break. A link is considered likely to break when the power of the signal received over it is close to the minimum detectable power. Route repair is the responsibility of a source node after receiving a warning about the imminence of a link break on an active route to a destination. This mechanism has been applied to DSR; AODV is also considered, but only superficially.

Predictive Preemptive Ad-Hoc On Demand Routing Protocol [13] propose a predictive preemptive approach to route maintenance. Route maintenance is initiated when a link break is expected rather than waiting for the break to happen. To evaluate the approach, we have added it to the AODV routing protocol, and evaluated its impact on performance using detailed simulations. The simulation results show that the proposed approach can be expected to improve performance significantly.

AOMDV (Ad hoc On-demand Multipath Distance 'Vector routing) [5] is a sophisticated protocol which produces multiple routes with loop-free and link-disjoint properties. When an intermediate node receives copies of a RREQ packet, it compares a hop count field in a packet with the minimum hop count, called *advertised-hopcount*, stored in a routing table for previous RREQ packets. Only a packet with the minimum hop count is accepted to avoid routing loops. Furthermore, a *firsthop* field in a RREQ packet is then compared with the *firsthop*-list in a routing table. When a route with node-disjoint property (new *firsthop*) is found, a new reverse route is recorded. Destination returns RREP packets accordingly, and multiple routes with link-disjoint property are established at a source node. A problem of AOMDV is that several efficient routes may be missed due to strong restriction by the *firsthop* field. Another problem is lack of backup route maintenance that causes timeout expiration of backup routes.

3. AOMDV Overview

Ad Hoc On-Demand Multipath Distance Vector Routing Protocol (AOMDV) [5] is one of the most used Ad-Hoc

routing protocol. It is a reactive routing protocol based on DSDV. AOMDV is designed for networks with tens to thousands of mobile nodes.

The main idea in AOMDV is to compute multiple paths during route discovery. It is designed primarily for highly dynamic ad hoc networks where link failures and route breaks occur frequently. When single path on-demand routing protocol such as AODV is used in such networks, a new route discovery is needed in response to every route break. Each route discovery is associated with high overhead and latency. This inefficiency can be avoided by having multiple redundant paths available.

The AOMDV protocol has two main components:

1. A route update rule to establish and maintain *multiple loop-free* paths at each node.
2. A distributed protocol to find *node-disjoint* paths that is route discovery.
3. The Route Maintenance Strategy.

In AOMDV a new route discovery is needed only when all paths to the destination break. A main feature of the AOMDV protocol is the use of routing information already available in the underlying AODV protocol as much as possible. Thus little additional overhead is required for the computation of multiple paths.

3.1 Route Discovery

The route discovery process has two major phases: route request phase and route reply phase. The route discovery process will be initiated when a route is requested by a source node and there is no information about the route in its routing table. First, the source node generates an RREQ and then floods the packet to networks. The RREQ's are propagated to neighbors within the source's transmission range. They also broadcast the packets to their neighbors. The process is repeated until the destination receives the RREQ. When an intermediate node receives the RREQ, it performs the following process:

1. When an intermediate node receives the information of RREQ, either it sends the route reply if the node is destination, or it rebroadcasts the RREQ to its neighbors.
2. The node reads the information from the RREQ.

In order to transmit route reply packets to the source, the node builds a reverse path to the source. The node will insert the path to its multiple path lists. Otherwise, the node will ignore the path and discard the RREQ.

3.2 Route Maintenance

Link failures in ad hoc networks are caused by mobility, less received power, congestion, packet collisions, node failures, and so on. In the AOMDV protocol, the link layer feedback from IEEE 802.11 is utilized to detect link failures. If a node sends packets along the broken link the node receives a link layer feedback. When a node detects a link break, it broadcasts route error (RERR) packets to its neighbors. If a source node receives the RERR, remove every entry from its routing table that uses the broken link. Differing from single path routing protocols, the route error packets should contain the information not only about the broken primary path but also the broken backup paths. When

the source node receives the RERR's, it removes all broken routing entries and uses the shortest backup paths as primary paths. The source node initiates a route discovery process where all backup paths are broken.

4. Predictive Preemptive AOMDV

4.1 Protocol Assumptions

In this section we present the operation of predictive Preemptive Ad-Hoc On Demand Multipath Distance Vector routing protocol (PPAOMDV). The Predictive Preemptive AOMDV is ad-hoc Reactive routing protocol based on DSDV. Predictive preemptive AOMDV is an extension of AOMDV, The goal behind the proposed protocol is to provide efficient recovery from "route failure" in dynamic network. To achieve this goal in this approach the "route failure prediction technique at the time of route discovery, computes the "received power of the receiver node" to predict pre-emptively before the route fails.

In Ad Hoc networks route failure may occurs due to less received power, mobility, congestion and node failures. Predictive Preemptive AOMDV predicts pre-emptively the route failure that occurs with the less received power.

4.2 Route Discovery

Predictive Preemptive AOMDV uses the Route Discovery method to discover the multiple node-disjoint paths. The route discovery uses the 2 different phases. Route Discovery phase and Route Reply phase. The Route Request phase is used to discover the path it broadcasts the RREQ packets. When an intermediate node receives the RREQ it performs following operation.

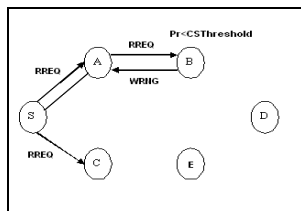


Figure 1. Route Request in EAOMDV

The above figure 1 shows the Route Request in EAOMDV; here the source node S wants to send the data to the destination D. Source S floods the RREQ packet to its neighbour nodes, Node A broadcasts the RREQ to the others nodes. After Broadcasting the Route Request Node B calculates the Receiver Power if it has less than Threshold it Drops the packet and sends the WRNG packet.

When an intermediate node receives the RREQ it performs following operations.

Step 1: The node measures its Received Power (Pr) by comparing the "Carrier Sense Threshold".

Step 2: If Received power Pr of receiver node is less than threshold then the host drop the packet and send the warning packet.

Step 3: The routes with the less received power can be avoided, and then selects another path for transmission. The received power is goes on checking at every node.

The following algorithm 1 shows the Route update Rule for Route Request:

Route Reply Phase, when the destination receives the route request packet, it sends route reply (RREP) packet to the source along the reverse paths created previously. The destination sends RREP to next nodes of reverse paths. They also forward the packet to next nodes until the source receives the RREP.

The pseudo code of our RREQ method is presented below; each time when a Route Request (RREQ) is received we execute the following code

Definitions:

RREQ: A route request packet.

CSthreshold: carrier sense threshold.

Pr: Received power.

WRNG: Warning Packet.

Procedure Route Request (RREQ)

Begin

```

When an intermediate node receives RREQ
if (Pr < CSthreshold) then
Drop the packet and send the WRNG packet to
source node.
else if (Pr == CSthreshold)
else ((Pr > CSthreshold) then
broadcast the RREQ to its neighbor nodes;
endif
else
Drop the packet RREQ and choose the another
path;
endif
    
```

Algorithm1. Route update rule for Route Request.

4.2 Data Transmission Phase

Predictive preemptive AOMDV sends the Route reply using the reverse path. When source node receives the RREP it starts the transferring the DATA packet from Source to Destination using the same path (forward path). Again after transferring the DATA packet it checks the Received power (Pr) of receiver node with the threshold value. If the pr is greater than the CSthreshold it transfers the DATA packet else drops the packet along with this path and sends the RERR packet. The received power is goes on checking at every node.

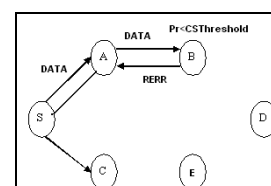


Figure 2. DATA Transmission in EAOMDV

The above figure 2 shows the DATA Transmission phase in Predictive preemptive AOMDV; here the source node S wants to send the DATA packet to the destination D. When an intermediate node receives the DATA packet, it performs following Steps.

Step 1: The node measures its Received Power (P_r) by comparing the “Carrier Sense Threshold”.

Step 2: If Received power P_r of receiver node is less than threshold then the host drop the packet and send the REER packet through the MAC Layer.

Step 3: The routes with the less received power can be avoided, and then selects another path for transmission. The received power is goes on checking at every node.

The following algorithm 2 shows the Route update Rule for DATA Transmission:

Definitions:

RERR: route Error.

CSThreshold: carrier sense threshold.

Pr: Received power.

DATA: DATA Packet.

Procedure Route Request (RREQ) update

Begin

When an intermediate node receives DATA
if ($P_r < CSThreshold$) then
Drops the packet and send the RERR packet to
source node.
else if ($P_r == CSThreshold$)
if ($(P_r > CSThreshold)$) then
boardcast the DATA to its neighbor nodes;
endif
else
Drop the DATA packet and sends the REER packet
and choose the another path;
endif

End

Algorithm2. Route update rule for DATA transmission.

Link failure is detected by the link layer and sends the REER packet through the MAC Layer. It receives the RRER the Route Maintenance phase is called to maintain the REER.

5. Simulation and Performance Results

5.1 Simulation Environment

The Predictive Preemptive AOMDV was evaluated using the Ns-2 [9] simulator version 2.32 with CMU's multihop wireless extensions. In the simulation, the IEEE 802.11 distributed coordination function was used as the medium access control protocol. The physical radio characteristics of each wireless host were based on Lucent's WaveLan.

WaveLan was direct spread spectrum radio and the channel had radio propagation range of 250 meters and capacity of 2Mb/sec. The AOMDV and Predictive Preemptive AOMDV (PPAOMDV) are to be compared in the simulation. Our results are based on the simulation of 50 wireless nodes forming an ad hoc network moving about in an area of 1500 meters by 300 meters for 100 seconds of simulated time. Nodes move according to the random waypoint model in a free space model.

The traffic patterns consists of 30 constant bit rate(CBR) sources sending 512 byte packets at a constant rate 4 packets per second. The random waypoint model was used to perform node movement. The movement scenario files used for each simulation are characterized by a pause time. Each node begins the simulation by selecting a random destination in the simulation area and moving to that destination at a speed distributed uniformly between 0 and 20 meters per second. It then remains stationary for pause time seconds. This scenario is repeated for the duration of the simulation. We carry out simulations with movement patterns generated for 5 different pause times: like 0, 10, 20, 30, 40 and 100 seconds. A pause time of 0 seconds corresponds to continuous motion, and a pause time of 100 (the length of the simulation) corresponds to limited motion.

5.2 Simulation Results and Analysis

The following performance metrics used to evaluate the two routing protocols:

Packet delivery ratio: The ratio of the data packets delivered to the destination to those generated by the CBR sources.

End to End delay: Average time between data packets received by the destinations and data packet sent by CBR source. The data were collected only successfully delivered packets. The delay is determined by any factors such as buffering during route discovery, queuing at the interface queue and routing paths between two nodes.

Overhead: The number of routing packets transmitted per data packet delivered to the destination.

Throughput: the total size of data packets that are received in CBR destinations per second. It represents in whether the protocols make good use of network resources or not.

We report the results of the simulation experiments for the original AOMDV protocol with the PPAOMDV. In this we analyze the performance metrics by the pause time.

Figure 3 compares the Average end-to-end delay by the different pause time. The routing protocols in varying in pause time. The graph demonstrates the node-disjoint PPAOMDV performs better than AOMDV; End-to-End delay is less Because of less route discovery.

Figure 4 plots the routing overhead of two routing protocols against pause time. Observe that node-disjoint PPAOMDV has a less overhead than AOMDV. The reasons for less overhead is less route discoveries are initiated in PPAOMDV, which lead to the flooding of RREQ.

Figure 5 compares the packet delivery ratio of routing protocols in varying pause time. In the simulation all the nodes move the same specified speed. The graph demonstrates the node-disjoint PPAOMDV performs better

then AOMDV; the paths in the Nod-Disjoint PPAOMDV fail independently due to their node-disjoint property.

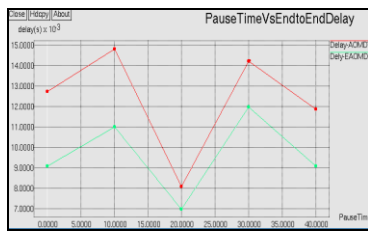


Figure 3. End-to-End Delay of AOMDV and PPAOMDV

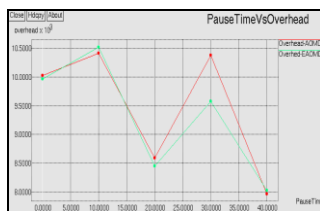


Figure 4: Overhead of AOMDV and PPAOMDV

Figure 6 compares the throughput for both the protocols. Throughput of PPAOMDV is better compared to AOMDV because of less Route Discovery; it saves the bandwidth and the network resources

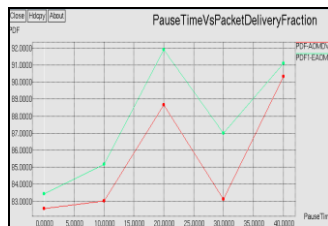


Figure 5: Packet Delivery Fraction of AOMDV and PPAOMDV

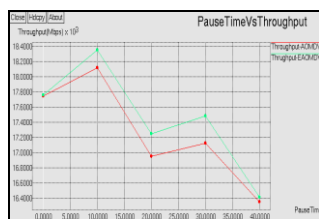


Figure 6: Throughput of AOMDV and PPAOMDV

6. Conclusion and Future Work

Multipath routing can be used in on-demand protocols to achieve faster and efficient recovery from route failures in highly dynamic Ad-hoc networks. In this project we have proposed an Enhanced Ad-Hoc on-Demand Multipath Distance Vector routing protocol that extends the AOMDV protocol to compute the stable path. There are three main contributions in this work. One is computing multiple loop-free paths at each node, another calculating received power of each node while transferring packets and node-disjoint route discovery mechanism.

Simulation results show that the throughputs, packet delivery fraction PPAOMDV are more than that of AOMDV. Also overhead of PPAOMDV is less than that of AOMDV. This is because PPAOMDV selects the node disjoint path pre-emptively before the path fails. If one path breaks then it selects an alternative and reliable node-Disjoint path. The advantage is less Route Discovery and Reduces the Route Error Packets.

In PPAOMDV it uses multiple paths only one path is used at a time. It is possible to use multiple paths simultaneously for increasing data rate, which is not considered in this project. This aspect can be one area for future work.

References

- [1] Abolhasan, M., Wysocki, T., and Dutkiewicz, E., "A review of routing protocols for mobile ad hoc networks," Ad Hoc Networks 2, pp. 1-22 (2004).
- [2] Perkins, C., Belding-Royer, E., and Das, S., "Ad hoc On-Demand Distance Vector (AODV) Routing," rfc3561.txt (2003).
- [3] Lee, S. J. and Gerla, M., "AODV-BR: Backup Routing in Ad hoc Networks," Proc. of IEEE Wireless Communications and Networking Conference, pp. 1311-1316 (2000).
- [4] Li, X. F. and Cuthbert, L., "On-demand Node-disjoint Multipath Routing in Wireless Ad hoc Networks," Proc. of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04).
- [5] Marina, M. K. and Das, S. R., "On-demand Multipath Distance Vector Routing for Ad Hoc Networks," Proc. of 9th IEEE Int. Conf. On Network Protocols, pp.14-23 (2001).
- [6] Jiang, M. H. and Jan, R. H., "An Efficient Multiple Paths Routing Protocol for Ad-hoc Networks," Proc. of the 15th International Conference on Information Networking, pp. 544-549 (2001).
- [7] DSR, "Dynamic Source Routing Protocol", IETF MANET Working Group Inter Draft
- [8] Theodore S. Rappaport, Wireless Communications Principles and Practice, Prentice Hall, December, 2001
- [9] The VINT Project. Network Simulator. <http://www.isi.edu/nsnam/ns/>.
- [10] Elizabeth M. Royer and C-K Toh: A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks.
- [11] Stephen Mueller, Rose P. Tsang, and Dipak Ghosal, Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges.
- [12] T. Goff et al., "Preemptive routing in ad hoc networks", *Journal of Parallel and Distributed Computing*, 2003, pp. 123-140
- [13] Sofiane Boukli Hacene et al, "Predictive Preemptive AODV", *Malaysian journal of computer science*.

Author Biographies

Sujata .

Mallapur Completed pre high school and high school in Chetan School in Gulbarga. B.E and M.tech in Poojya Doddappa Appa college of Engineering in the year of 2009. Presently working as lecturer in Appa Institute of Engineering And Technology, Gulbarga. My Instrested Area is Ad-Hoc Network.

A Novel Design of Multi-Port Cartesian Router

R.Anitha^{#1}, Dr.P.Renuga^{#2}

^{#1}Lecturer, Department of Electronics and Communication Engineering, PSNACET, Dindigul
¹anitha_rvs@yahoo.co.in

^{#2}Assistant Professor, Department of Electrical and Electronics Engineering, TCE, Madurai
²preee@tce.edu

Abstract: - Network-on-chip (NoC) architectures are emerging for the highly scalable, reliable, and modular on-chip communication infrastructure platform. The NoC architecture uses layered protocols and packet-switched networks which consist of on-chip routers, links, and network interfaces on a predefined topology. In this Paper we design network-on-chip which is based on the Cartesian network environment. This paper proposes the new Cartesian topology which is used to reduce network routing time, and it is a suitable alternate to network design and implementation.

The Cartesian Network-On-Chip can be modeled using Verilog HDL and simulated using Modelsim software.

Keywords: - Cartesian Networks, NoC, Routing algorithm, Multi-port Cartesian Router

I. Introduction

Cartesian routing is a fast packet routing mechanism intended for geographic addresses and can effectively accelerate the packet routing process within a local or metropolitan environment. The wide area Cartesian routing described in this paper is an extension of the Cartesian routing algorithms designed to make the exchange of internet work packets between geographical regions possible.. The proposed Internet is viewed as a hierarchy of networks consisting of routers. At the highest level of this hierarchy, major routers exchange packets between large geopolitical areas such as countries, states, or provinces. At the lowest level of the structure, packets are routed between local routers in small geographical regions ranging from an office to a small town The Cartesian Routing algorithm overcomes these problems by creating a hierarchical network consisting of two or more layers. Each network at a given layer encompasses one or more networks Each network, regardless of its layer, employs the Cartesian Routing algorithm for packet routing. Two extensions to the original Cartesian Routing algorithm are required: each network (except for the highest) requires an internet work router that can direct

packets destined for other networks “up” to the encompassing network. The address structure reflects the network structure, with specific fields in the address associated with each layer.

II. Cartesian Networks

A Cartesian network consists of a set of *collectors* and one or more *arterials*

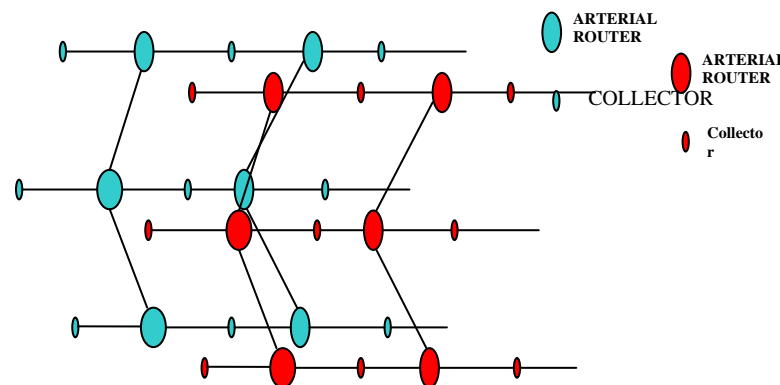


Fig.1.Cartesian Router

Each collector is a chain of *collector routers* running east-west and sharing common latitude. Collector routers have two side ports (east and west) to exchange packets' horizontally'. Each collector router also has a *bottom* Port which allows it to connect to a set of local hosts. Arterials exchange packets between collectors. Each arterial router, except the most northerly and the most southerly has, at least, four ports (north, south, east and west). Arterials need not share a common longitude. In a Cartesian network, the imposed topological structure relieves each router from maintaining routing tables. Each router is bound to a unique pair of addresses, the state information is minimal, and each router maintains the accessibility of arterials to its west and east.

2.1 Cartesian Network Initialization

In Cartesian routing, each arterial issue *Arterial This Way* (ATW) control packets during its initialization process. An ATW tells the receiving collector router if an arterial is accessible through the incoming port. An ATW also specifies what kind of connection is accessible via the incoming port: north, south, north and south or neither. Upon receiving an ATW, each collector router updates its *Arterial Direction Indicator* (ADI) and forwards the ATW to the opposite port. ATWs are also used to establish *virtual Arterials*, constructed in situations where it is physically impossible for an arterial to span two collectors. The ADI points in the direction of the arterial router (i.e., east or west) and indicates whether the arterial router has a connection to the north, the south, or both. Figure 1 illustrates a Cartesian network.

Packets can arrive on either a west or east port of a collector router. Packets intended for different latitude are forwarded out the opposite port from which they are received. The ADI determines the packet's initial direction on the collector router when a packet arrives on the bottom port collector router. In deciding a packet's initial direction, the router first compares the packet's destination address with its own address. The packet will be forwarded in the direction of the destination if the destination latitude is the same as the collectors. The packet is forwarded in the direction of the ADI if the destination is on different latitude.

2.2 4 Port Cartesian Routing Algorithm

2.1.2 Collector Router

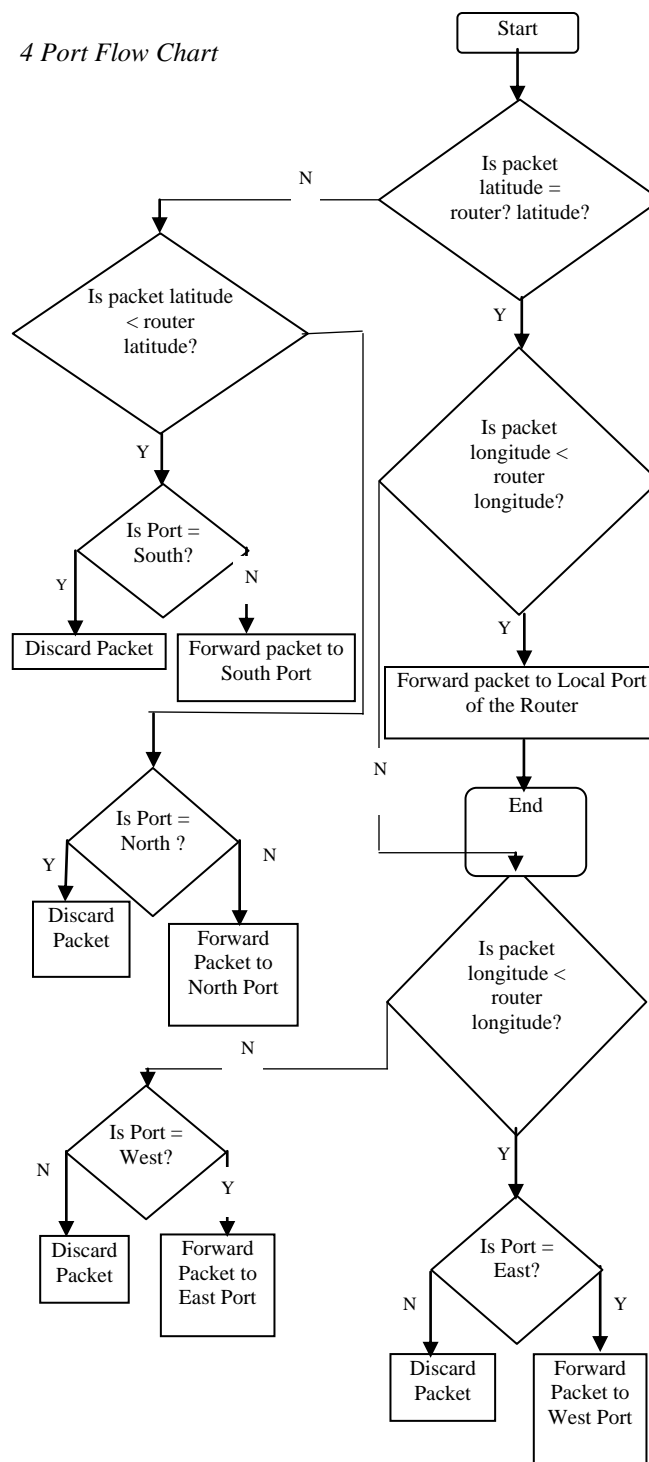
The Collector Router is responsible for routing a unicast packet on 'horizontal', collector networks. This router has tree ports: West, East and Local. When a packet arrives at a port; its longitude address is compared with the router's longitude address. If the packet's longitude is greater than that of router, the packet is forwarded to the east port. If less than, the packet is sent towards west, and if equal the packet belongs to the local port. The packet is discarded when a packet is determined to have a non-existent Cartesian address; this is to prevent packet circulation.

2.1.1 Arterial Routers

The Arterial router is responsible for routing packets from one collector network to another. An arterial router has four ports: North, East, South and West. When a packet arrives at an arterial router, its latitude is compared with the router's latitude.

If they are not equal, it implies that the packet either belongs to an 'upper' or 'lower' collector network; if packet's latitude is greater, the packet is forwarded north otherwise it is forwarded south. If the packet to west or east. Arterial routers also discard packets with addresses of non-existent destinations.

4 Port Flow Chart



ADDRESS	FUNCTIONS
Destination latitude=node_latitude	Router keeps the Packet
Destination_longitude=node_longitude	Packet routed to North Port
Destination_latitude>node_latitude	Packet routed to South Port
Destination_longitude>node_longitude	Packet routed to East Port
Destination_latitude<node_latitude	Packet Discarded

Process flow for 4 Port Cartesian Network

Table 1:4-port Routing algorithm

The Routing algorithm for the Cartesian Network is if the Destination Latitude is equal to the node latitude and also the Destination longitude is equal to the node longitude means the router hold the packet. If the destination Latitude is greater than load latitude and also destination longitudes greater than node longitude means the packet routed to north port. If the destination latitude lesser than load latitude and also destination longitudes greater than node longitude means the packet routed to south port .If the destination latitude is equal to the node latitude and also destination longitude is greater than node longitude means the packet routed to East port. If the destination

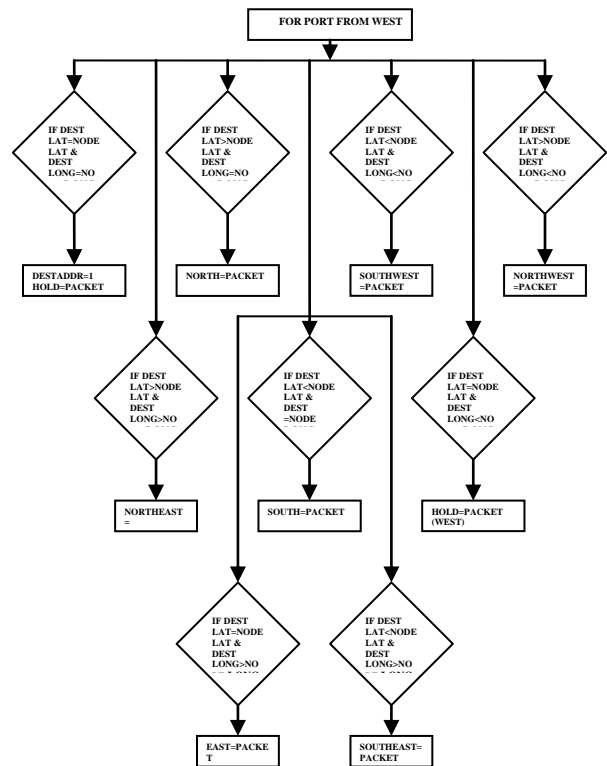
ADDRESS	FUNCTIONS
IF DEST LAT=NODE LAT & DEST LONG=NODE LONG	Router keeps the Packet
IF DEST LAT>NODE LAT & DEST LONG=NODE LONG	Packet routed to North Port
IF DEST LAT<NODE LAT & DEST LONG<NODE LONG	Packet routed to South West Port
IF DEST LAT>NODE LAT & DEST LONG<NODE LONG	Packet routed to North Port
IF DEST LAT>NODE LAT & DEST LONG>NODE LONG	Packet routed North East
IF DEST LAT<NODE LAT & DEST LONG=NODE LONG	Packet routed to South port
IF DEST LAT=NODE LAT & DEST LONG<NODE LONG	Router hold the Packet (WEST)
IF DEST LAT=NODE LAT & DEST LONG>NODE LONG	Packet routed to East port
IF DEST LAT<NODE LAT & DEST LONG>NODE LONG	Packet routed to South East port

longitude is lesser than node longitude means the packet will discarded

Table 2: 8 Port Routing algorithms

If destination latitude is equal to node latitude and destination longitude is equal to node longitude then router keeps the packet. If destination latitude is greater than to node latitude and destination longitude is equal to node longitude then packets routed to north port.

8-Port Flowchart:



2.3 Wide Areas Cartesian Networks

A Cartesian network provides a straightforward topological structure that relieves collector routers from the need to maintain routing tables. However, it would be unrealistic to implement a single worldwide Cartesian network. Such a widespread Cartesian network, for example, requires every packet destined for a router with the same latitude identifier as the source router's latitude identifier to visit all the collector routers. It is also necessary for such a network to have one collector for every possible latitude. These limitations suggest that implementing a single worldwide Cartesian network would be impractical. An alternative to a worldwide Cartesian network is to create a set of smaller Cartesian networks and implement a mechanism for interchanging packets between them. One

approach to interchanging packets between Cartesian networks is to forward packets towards their destinations.

When a packet reaches the boundary of a network it “falls off” the edge and is delivered to a special router to be forwarded towards the destination address. The process of routing a packet from one network to another using this approach becomes problematic when networks are *interleaved* or *overlapped*. Two networks are considered interleaved if there is at least one collector router on one of the networks where its longitude identifier lies between the longitude identifiers of two collectors from the other network and its latitude identifier lies between the latitude identifiers of two collectors from the other network. Figure illustrates two interleaved networks. Two networks are said to be overlapped if there is at least one collector router on one of the networks where its longitude identifier lies between the longitude identifiers of two collectors from the other network and all three of them share the same latitude identifier. Figure illustrates two overlapped networks.

An alternative method for delivering a packet to its destination is to find the destination network address and then to route the packet to the destination network by using Cartesian routing algorithms. This implies that each network must be identifiable using the packet’s destination address. If we assume that each network has a rectangular shape, recognizing the destination network is a matter of comparing the packet’s destination address with the network’s boundaries. However, there are a number of reasons to assume that it would be unrealistic to expect networks to have rectangular borders: geographical barriers and political jurisdictions, for example. Since Cartesian routing uses latitude and longitude pairs to identify the source and the destination addresses of packets, this information is not sufficient to determine to which network a collector/arterial belongs in the case of interleaved and overlapped networks. This, in turn, suggests that an additional set of information is required to identify to which network a collector or arterial is connected. To achieve this, the authors propose a hierarchical structure for Cartesian networks. In the next section the possibility of multiple-layer Cartesian networks as a solution for interchanging packets between arbitrary shaped interleaved and overlapped Cartesian networks are explained. In the remainder of this paper, the terms “wide area Cartesian networks” and “multiple-layer Cartesian networks” are used interchangeably.

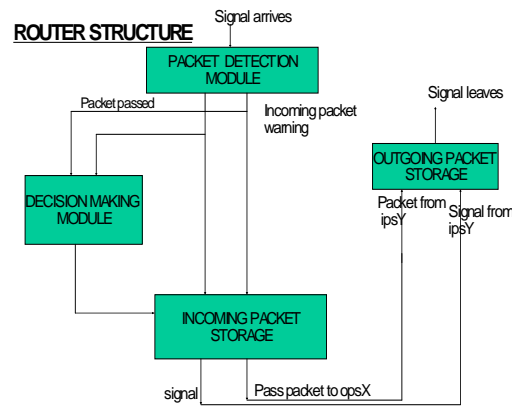


Fig 2: Router Structure

III. MULTIPLE-LAYER CARTESIAN NETWORKS

Multiple-layer Cartesian networks impose a new set of topological dependencies among a set of Cartesian networks, such that interchanging packets between networks is feasible without creating and maintaining routing tables. Generally, in multiple layer Cartesian networks, the idea of Cartesian networks is expanded in a larger scale using a hierarchical structure.

3.1 DMM

Multiple-layer Cartesian networks have a hierarchical structure. The highest layer of the hierarchy is a single Cartesian network. Each underlying layer consists of a set of mutually disjoint Cartesian networks (i.e., they are physically disjoint and share no collector or arterial router); however, networks in the same layer can be interleaved or overlapped.

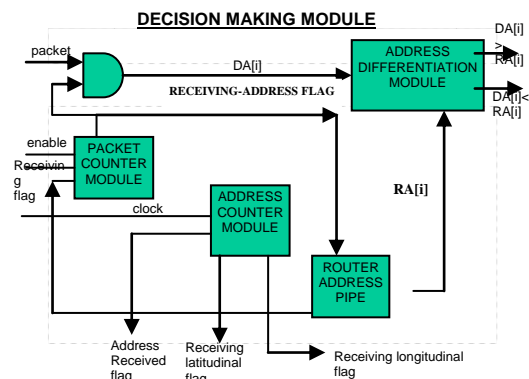


Fig 3: Decision Making Module

The PCM (Packet Counter Module) strips the destination address from the packet. It counts the incoming

packet bits and sets the RECEIVING-ADDRESS flag when the first bit of the address is read. The ACM (Address Counter Module) keeps the track of the number of address bits that have been read. It indicates which portion of the address is being received (latitude or longitude) and if the entire address is received, it sets the ADDRESS-RECEIVED flag. When the ADDRESS-RECEIVED flag is set, the IPS is told to keep the packet. The RAP stores the router address and pipes it out serially so that it can be compared with the incoming destination address.

The Router address is static value, and is kept in non-volatile memory for long term storage. The ADM simply compares the *i*-th bit of the destination address to the *i*-th bit of the router address as the destination address is read in to the DMM. If the DA does not equal to RA, the ADM tells the incoming packet storage (IPS) module to destroy the stored packet.

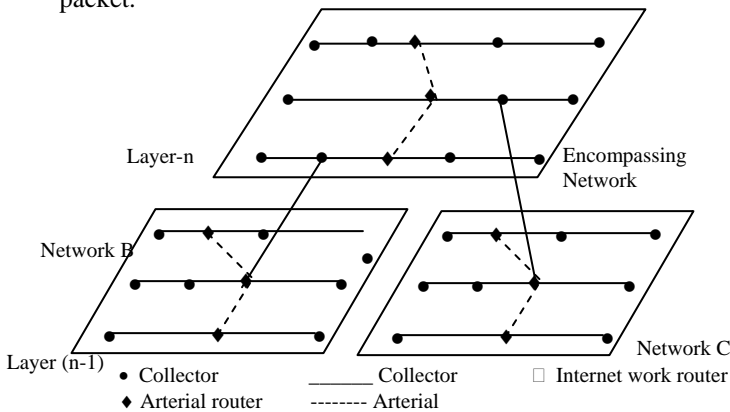


Figure 4: An encompassing network (A) and two encompassed networks (B and C)

3.2 Router Identification

In a Cartesian network, each router is bound to a Cartesian address. Whereas, in a multiple-layer Cartesian network, each router is bound to an *identifier*. The identifier of a router at layer-*m* of an *m*-layer Cartesian network is the same as its Cartesian address. An identifier of a router at lower layers is the identifier of its immediate encompassing router, followed by the Cartesian address of the router itself, meaning that an identifier is an ordered list of Cartesian addresses. For example, routers at layer-*(m-1)* maintain the Cartesian address of the router that represents the network to which they belong, followed by their own Cartesian address. In general, in an *m*-layer Cartesian network, each router at layer-*n* maintains a list of *(m-n+1)* ordered Cartesian addresses, where. For example, routers at the lowest layer of the hierarchy, layer-1, which are connected to local hosts through their bottom ports, are bound to *m* ordered Cartesian addresses: *m-1* correspond to the identifier of the encompassing router at layer-2, and one, the Cartesian

address of the router itself. Figure 5 illustrates the hierarchical addressing structure of an identifier for a router at layer-*n* of an *m*-layer Cartesian network. The hierarchical addressing structure overcomes the problems with both interleaved and overlapped networks since every router in the hierarchy has a unique address. It also enables a router in a network at layer to determine if a packet is local to the network or not. A packet is said to be *local* to a network if the network encompasses the destination address of the packet. A router can determine this by comparing the most significant *m-n* Cartesian addresses of the packet's destination address with the first *m-n* Cartesian addresses of its own identifier.

PACKET FORMAT

- SOP → Start to Packet (6 bits---6'b111111)
- EOP → End of Packet (6 bits---6'b111111)
- M → Cast (Unicast - 4'b0000, Multicast -b'0001)
- DLAT → Destination Latitude
- DLON → Destination Longitude

SOP	M	DLAT	DLON	SLAT	SLONG	DATA	EOP
-----	---	------	------	------	-------	------	-----

3.3 Packet Routing

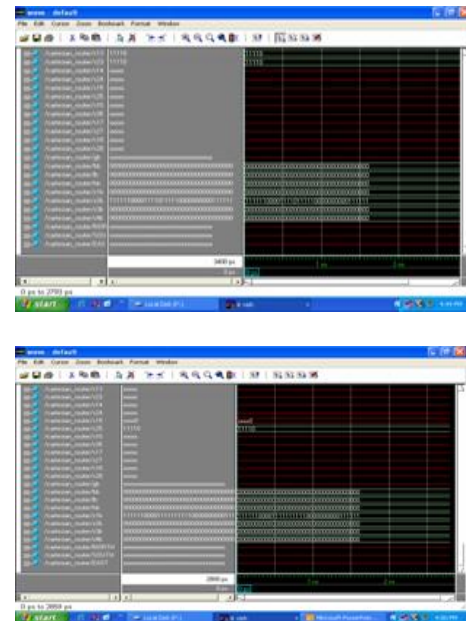
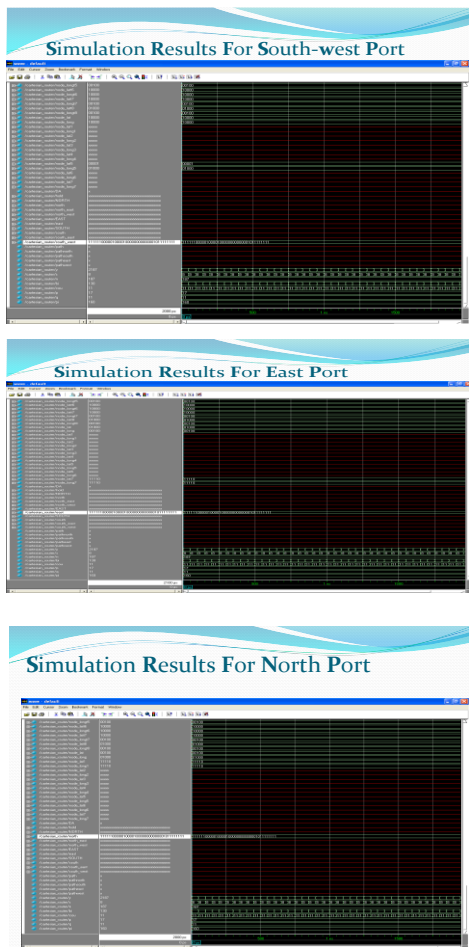
In an *m*-layer Cartesian network, each collector router at layer-*n* is bound to an identifier which is a list of *m-n+1* Cartesian addresses. A packet can enter a network at layer-*n* through the bottom port of a collector router or the top port of the network's IR. Packets received on the bottom port of a collector router are either local or non-local to the network, as described above. When a packet is found to be local, that is, the network encompasses the destination address, the router "tags" the packet as a local packet by setting a single bit of the packet's address called the *local bit*. When a packet is local to a network at layer-*n*, the *(m-n+1)* th Cartesian address of the packet's destination address is used to route the packet in the network using Cartesian routing algorithm. For example, at layer-*m*, the first Cartesian address is used for Cartesian routing, while at layer-1 the *m*th address is used. When a packet is received by a router on its bottom port, the address is Cartesian address of the encompassing router at layer-*(m-1)* inspected. Non-local packets must be sent towards the network's IR in order to be delivered to the encompassing network. The router clears the local bit and forwards the packet towards the IR. Forwarding a packet to the IR requires that each collector router and arterial maintains an *Internet work Router Direction Indicator* or IRDI. The IRDI determines if the internet work router is accessible through the west port, east port or neither, in the case of collector routers; or whether it is accessible through the north port, south port or neither, in the case of arterials. If the packet is determined to be non-local, it is forwarded in the direction specified by the IRDI.

When the IRDI indicates that IR is not accessible, the packet is dropped and a message is returned to the source notifying that the destination address is not reachable.

A collector router that receives a packet on its west or east port checks the local bit; if set the Cartesian routing algorithm is employed to route the packet, otherwise the packet is forwarded to the opposite port. When a packet enters a network through the top port of the network's IR, the packet is guaranteed to be local, since this has already been verified by the encompassing network. Upon receiving the packet, the IR sets the local bit and then applies the Cartesian routing algorithm on the $(m-n+1)$ th Cartesian address of the packet's destination address.

IV.SIMULATION RESULTS

4.1. Router Design simulation Result



V.CONCLUSION

In Previous work, they designed conventional router which uses a routing table to determine whether to keep, forward or discard the packets. As networks grow in size, the memory requirements of the routing tables increases proportionally. The average search time increases as the routing table increases. ASIC based router design.

In our Proposed work, the New Cartesian network is designed, which is Independent of routing table. It is the High speed network transmission when compared with existing work.

VI.REFERENCES

- [1].Low-Power Network-on-Chip for High-Performance SoC Design Kingman Lee, *Student Member, IEEE*, Se-Joong Lee, *Member, IEEE*, and Hoi-Jun Yoo, *Senior Member, IEEE* IEEE transactions On Very Large Scale Integration (VLSI) Systems," VOL. 14, NO. 2, February 2006.
- [2].W. Dally et al., "Route packets, not wires: On-chip interconnection networks," in Proc. Des. Autom. Conf., Jun. 2001, pp. 684–689.
- [3].L. Benini et al., "Networks on chips: A new SoC paradigm," IEEE Computer, vol. 36, no. 1, pp. 70–78, Jan. 2002.
- [4].D. Bertozzi et al., "Xpipes: A network-on-chip architecture for gig scale system-on-chip," IEEE Circuits Syst. Mag., vol. 4, no. 2, pp. 18–31, 2004.
- [5].E. Rijpkema et al., "Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip," in Proc. Des., Autom. Test Europe Conf., Mar. 2003, pp. 350–355.
- [6].V. Nollet et al., "Operating-system controlled network on chip," in Proc. Des. Autom. Conf., Jun. 2004, pp. 256–259.
- [1] A. Ivanov and G.D. Michele, "The Network-on-Chip Paradigm in Practice and Research," IEEE Design and Test of Computers, vol. 22, no. 5, pp. 399-403, Sept.-Oct. 2005.

- [2] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Milberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A Network on Chip Architecture and Design Methodology," Proc. IEEE CS Ann. Symp. VLSI, p. 117, 2002.
- [3] W.J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," Proc. 38th Design Automation Conf. (DAC '01), pp. 684-689, 2001.
- [4] F. Karim, A. Nguyen, and S. Dey, "An Interconnect Architecture for Networking Systems on Chips," IEEE Micro, vol. 22, no. 5, pp. 36-45, Sept.-Oct. 2002.
- [5] P.P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a Switch for Network on Chip Applications," Proc. IEEE Int'l Symp. Circuits and Systems (ISCAS '03), vol. 5, pp. 217-220, May 2003.
- [6] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip," ACM Computing Surveys, vol. 38, no. 1, pp. 1-51, 2006.
- [7] P.P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance Evaluation and Design Trade-Offs for Networkon- Chip Interconnect Architectures," IEEE Trans. Computers, vol. 54, no. 8, pp. 1025-1040, Aug. 2005.
- [8] D. Linder and J. Harden, "An Adaptive and Fault-Tolerant Wormhole Routing Strategy for k-Ary n-Cubes," IEEE Trans. Computers, vol. 40, no. 1, pp. 2-12, Jan. 1991.

Ad-hoc Networking Applications In Different Scenarios

Md. Taslim Arefin

Department of Electronics and Telecommunication Engineering
Faculty of Science and Information Technology
Daffodil International University
Dhaka, Bangladesh
Email: arefin@daffodilvarsity.edu.bd

Abstract- This paper presents a significant aspect of Ad-hoc networking application based on different scenarios. In the wireless communication systems, there is a need for the rapid deployment of independent mobile users. Significant examples include establishing survivable, efficient, dynamic communication for emergency operations, disaster relief efforts, and military networks. Such network scenarios cannot rely on centralized and organized connectivity. This can be conceived as applications of MANET (Mobile Ad-Hoc Network). Most of the concerns of interest to MANETs are of interest in VANETs (Vehicular Ad-Hoc Network). Cellular ad-hoc networks also seem to be a promising solution for broadband wireless access networks in beyond 3G systems. This paper also describes the prominent future of ad-hoc networking in future wireless communication system.

Keywords – Ad-hoc network, MANET, VANET, Cellular Ad-hoc networks.

I. INTRODUCTION

This paper describes different types of ad-hoc network application based on its different scenarios. The vision with a MANET is a spontaneous network that can be established even if the local infrastructure has been destroyed. It can be used in a natural disaster area, military operation and also for educational, business purposes. VANET is used to provide communications among the nearby vehicles and between nearby fixed equipment such as traffic signal, road side alarm. In this paper we will discuss about it. This paper also discusses the role of ad-hoc networking in future wireless communications indicating that cellular ad-hoc networking seems to be a promising solution to fulfill the requirements of future wireless communication systems.

II. AD-HOC NETWORKING IN DISASTER AREA

In recent years, large scale sizes of natural disasters, such as earth quake, mountain explosion, hurricane, rain flooding and snow-slide are occurring in many countries in the world frequently. Many people by those disasters are losing their lives; huge amount of properties is being destroyed. When a disaster happens in an area where people live in and there are victims at there, rescue teams are organized and sent to save the victims. However, in many cases of disaster occurrences, the resident lives can be saved if the disaster information network system could effectively work just after disaster happened.

In typical disaster scenario emergency service teams, and relief agencies quickly mobilize to aid those affected by the disaster. The disaster zone can be large and geographically spread apart. Unfortunately, relief operations are often hampered due to communication system failure. Failure to communicate accurate and timely information can cost lives of the victims as well as those who are trying to assist. Under these circumstances, a reliable communications infrastructure that provides the required information in a timely manner can solve the problem.

Network establishment

In a disaster area where all communication system is fully or partly damaged, there must be an establishment of a network for relief and rescue mission. Because large disasters destroy core terrestrial communications infrastructure, and backup networks are often unable to handle the necessary traffic volumes. Relief and rescue teams may need to communicate with either the central command center, or with other teams on the disaster site.

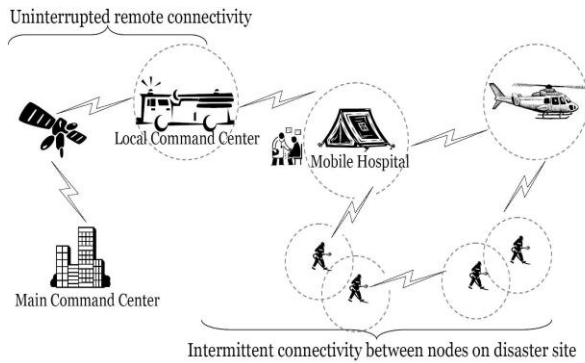


Fig: 1 A typical information sharing system in a disaster area

The Fig. 1[2] shows a typical information sharing system in a disaster area. In the first scenario, the long-distance communication link with the main command center is typically established through VSAT terminals that are mounted at selected mobile command centers. The intermittent connectivity between the mobile nodes is generated by a wireless ad-hoc network, typically a mobile ad-hoc network (MANET).

Challenges

To establish information sharing system for disaster response some requirements and challenges should be considered. Interoperability is one key problem that can be addressed through the use of cognitive radios. We list further requirements and challenges below [2]

- **Rapid deployment:** The communications infrastructure needs to be rapidly deployable. As such, it should be self organizing, and discover other nodes and communication opportunities without requiring manual intervention. This requires the capability to correctly infer the current context, determine the best sequence of actions, and then react accordingly.

- **Adaptable:** Network and communication characteristics can vary considerably over the course of the disaster relief operation, and thus cannot be pre-determined. This includes characteristics such as the network topology, type and size of exchanged traffic, as well as changing application service requirements, e.g., the type of data collected can range from aggregate statistics on casualty numbers, to images/video content of a specific site. This necessitates that the communication infrastructure be flexible enough to accommodate these changing requirements without sacrificing system performance or hindering other operational aspects of the network.

- **Resilient and Robust:** The communication infrastructure needs to be resilient and robust. This is challenging as nodes can be highly mobile, resulting in frequent disconnections and link failures. An end-to-end connection may never exist between pairs of nodes wishing to share information. This imposes challenges on both the communication protocols as well as the communications infrastructure. The communication protocols need to be designed such that they are disruption-tolerant. Similarly, the communications infrastructure needs to be modified to

enable a store-and forward approach, whenever disconnections are encountered.

- **Incrementally Deployable:** Any feasible replacement for the current generation of disaster response communication systems must be incrementally deployable, so as not to require a complete overhauling of the existing infrastructure. It must also be backwards compatible, where new users can benefit from using the information sharing network, while also having the ability to communicate with legacy radio systems.

- **Power conservation:** End user devices must be small and portable, limiting their battery size. Power conservation is an important issue as teams of first responders may remain deployed at forward bases for days, or even weeks.

- **Security and privacy requirements:** Depending on application requirements, network security and data privacy requirements can vary significantly as well. e.g., when the network is used primarily for facilitating exchange of statistical information, data confidentiality (or anonymity) might not be particularly required, though maintaining data integrity might still be important. However, if the network is used to transmit medical files for patients undergoing treatment, then user authentication and end-to-end data confidentiality may be needed, as per governmental regulations (e.g., HIPAA requirements for security and privacy of health data in the U.S. **Error! Reference source not found.**). Balancing these security issues against the requirements for timely and efficient exchange of information in life-and-death situations involves both policy and technical issues that need to be understood.

III. AD-HOC NETWORKING IN MILITARY USE

Ad-hoc networking is an enabling technology relevant to vast number scenarios. It can be a fundamental key model for future networking of the armed force in both war and peace keeping operations. The military aspects in a mobile ad hoc network are especially interesting and a bit complicated. In a military scenario with a hostile environment where more things need to consider and also it are harder constraints than in a MANET for educational or business purposes. For example, a military scenario may have higher requirements regarding the information security.

Network requirement

Military networks are probably the most difficult ad hoc network to handle when it comes to mobility management and mobile communication. There are a number of things to take into concern as shown by Hannu H. Kari in [1]

- **Hostile enemy** – If the enemy can get the communication in the network to stop function properly or be able to tamper with the messages, the enemy can get great advantages.

- **Trust models** – How to deal with the level of trust and compromised nodes.

- **Quality of service control** – Not all nodes and packets are equal.

- Radio power usage restrictions – Battery, reveal location, time and importance of the node.
- The need for robustness – Fault resilience, redundant routes, automatic repair after failure.
- The need for performance

Military application

For military application, the first question comes to mind is that, can mobile ad-hoc network being a reliable group communication in a military operation? To find the solution Thales Research and Technology presented a number of applications of ad-hoc networking in 2004 [2]. So, according to the research the different types of applications of MANET can be used in military operation are as follows [2]:

Unmanned vehicles

Unmanned vehicles are set to play an increasing role in the future armed force’s operational missions. They can be used as individual platforms for surveillance or other purpose.

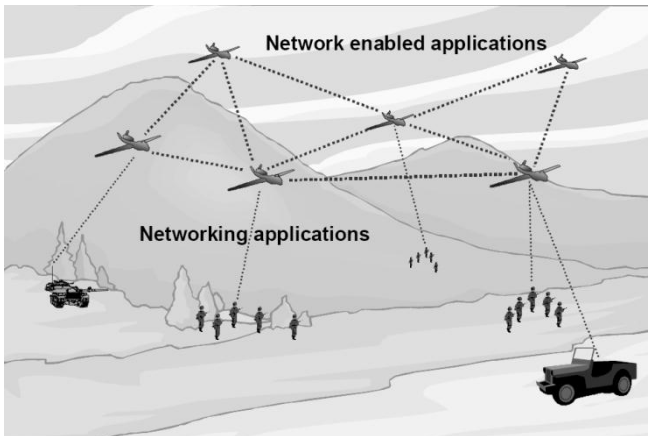


Fig. 2. Unmanned vehicles

Fig. 2 shows an example of unmanned vehicles application where a group of unmanned air vehicles (UAV) is deployed in a battle field. There are two types of applications supported by this network. Those are described below.

- Networking applications: The ad-hoc network shaped by the UAV in the sky can offer a backbone for land based stages to communicate when they are out of direct range or when obstacles prevent direct communication. The ad-hoc network therefore extends down to the land based forces and allows communication across battlefield.
- Network enabled application: the ad-hoc network also allows network enabled applications to run on the UAVs. For example, they may be able to enhance their performance by collaborating with each other over the network, for navigation, surveillance or combat purposes.

Sensor networks

Sensor networks produce a system which is more capable than the sum of its parts. There are three types of sensors are deployed in fig. 3 [2] are described below.

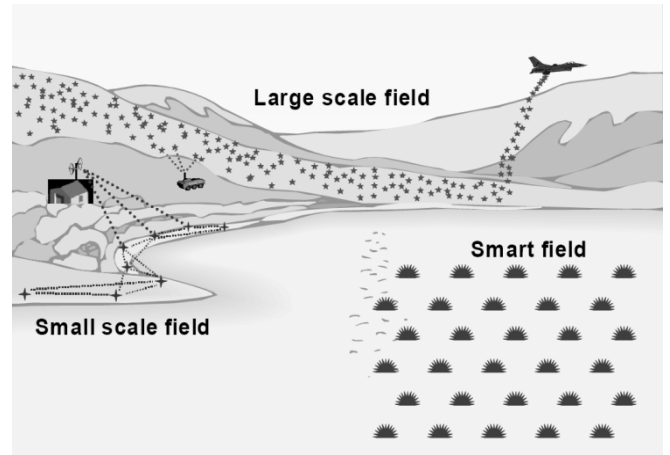


Fig. 3. Sensor networks

- Small scale fields: small scale sensor fields are deployed in strategic places. They would traditionally be individually configured and connected to a central controller. Ad-hoc technologies can simplify the deployment load by removing the entire network configuration. The sensors only need to be positioned so that they are in range as shown in fig. 3.
- Large Scale fields: large scale fields are shown in fig. 3 as being deployed from a plane, scattering the sensors over an area of interest below. The main characteristic of this sensor is the large number of sensors deployed, making any form of manual or individual configuration not feasible. Using ad-hoc network in this scenario allows the sensors to form a network among them where they land. The field will be linked, agree to sensing data to be composed from all parts of the network by a passing vehicle.

Mobile communication

Armed forces deployed in offensive or peace keeping operations need to communicate on the move, whether between vehicles or between dismounted troops. Fig. 4 shows a convoy on the left and a dismounted section on the right.

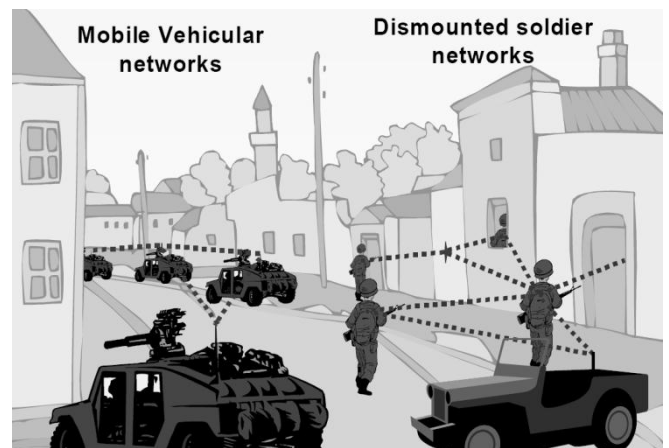


Fig. 4. Mobile Communication

There are two types of network in this system which is given bellow.

- **Mobile vehicular network:** on the move all the vehicles need to communicate frequently. In order to achieve all vehicles in a line the radio technology is required to have a range capable to cover the extent of the vehicle formation. Ad-hoc technologies calm down this constraint, by only requiring each vehicle to be in range of its nearby neighbors. The suitable neighbor will forward any data which is designed for a further distant vehicle. As the vehicle travel, the connectivity between them modify as they come in and out of collection of each other.

- **Dismount soldier network:** Correspondingly for the dismantled soldier, a dynamic network small range radio is enough to extent the entire section and provides communications beyond the each and every soldiers reach. A further feature for this is the exploit of committed relays which a soldier can leave behind him when he moves away from his section. This presents relays which strengthen the networks connectivity if the section is extend or operating in cruel condition such as pretending deep into a building.

IV. VANET APPLICATIONS

Automobile traffic is a key trouble in modern cities. Daily millions of hours and gallons of fuel are dissipated all over the world only by vehicles trapped in traffic jam. There was a study in 2005 by the Texas Transportation Institute (TTI) [4]. According to the 2005 Urban Mobility Report, traffic congestion is growing across the nation, costing Americans \$63.1 billion a year. The report measured traffic congestion trends from 1982 to 2003. A system that could deliver an accurate map of traffic to drivers in real time could save huge amounts of money. If such a system could be deployed cheaply it would be very profitable and decrease the environmental impact of automobile traffic. For this reason there is a growing commercial and research interest in the development and deployment of Vehicular Ad-Hoc Network (VANET).

VANET is a special case of Mobile Ad-Hoc Network (MANET). It consists of a number of vehicles traveling on urban streets and capable of communicating with each other without a fixed communication infrastructure. VANET is expected to be of great benefit for safety applications, gathering and disseminating real-time traffic congestion and routing information, sharing of wireless channel for mobile applications etc.

Factors influence mobility in VANET:

The mobility pattern of nodes in a VANET influences the route discovery, maintenance, reconstruction, consistency. A VANET can have both static (non-moving) and dynamic (moving) nodes at any instance. The static nodes tend to dampen the changes in topology and routing by acting as stable relaying points for packets to/from the neighboring nodes. On the other hand, dynamic nodes add entropy to the system and cause frequent route setups, teardowns, and packet losses. The effects of various factors that influence the mobility pattern in VANET [6] are as follows:

- **Layout of streets:** Streets force nodes to confine their movements to well defined paths. The layout of the streets might be such that vehicles travel on parallel streets

spaced far apart might be out of communication range. Streets can have either single or multiple lanes and can agree to either one way or two way traffic.

- **The Block size:** Urban areas are normally alienated into blocks of different sizes. A city block can be considered the nominal area surrounded by streets. These blocks can be of different sizes. Metropolitan areas generally have smaller city blocks than smaller towns. The block size states the density of the intersections in that particular area, which in revolve determines the frequency with which a vehicle bring to an ends. It also decides whether nodes at neighboring intersections can listen to each other's radio transmissions. Bigger blocks would enlarge the network's sensitivity to vehicles clustering at intersections and to network partitioning, and result in a mortified performance.

- **Traffic control mechanisms:** The most common traffic control mechanisms at intersections are stop signs and traffic lights. A vehicle requires stop at a red light until it turns green. A vehicle also needs to stop at a stop sign for a few seconds before moving onward. These mechanisms cause the formation of clusters and queues of vehicles at intersections, consequently reducing their average speed. Reduced mobility implies more static nodes and slower rates of route changes in the network. Alternatively, cluster arrangement can also adversely influence network performance with increased wireless channel argument and rise network partitioning.

- **Interdependent vehicular motion:** Vehicles cannot disregard physical constraints posed by the presence of streets and nearby vehicles. Every vehicle's movement is influenced by the movement pattern of its surrounding vehicles. For example, a vehicle needs to maintain a minimum safe distance from the one in front of it, increase or decrease its speed, or change to another lane to avoid congestion.

- **Average speed:** The speed of the vehicle determines how quickly its position changes, which in turn determines the rate of network topology change. The speed limit of each road determines the average speed of vehicles and how often the existing routes are broken or new routes are established. Additionally, vehicles' acceleration / deceleration and the map's topology also affect their average speed - if a map has fewer intersections, it implies that its roads are longer, allowing vehicles to move at higher speeds for longer periods of time.

VANET system design

VANET is designed to be a useful system to drivers. In this system design the driver can not only exchange information on every section of road but also it can exchange information between the vehicles on areas of unexpected

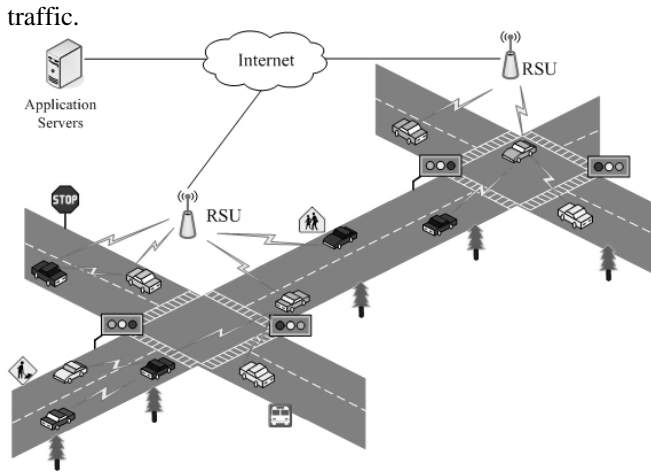


Fig. 5. An example of VANET architecture

Fig. 5 defines a simple VANET architecture. Vehicular Ad-hoc Networks (VANET) can be built by equipping vehicles with onboard sensing devices. Every corner of streets there is a RSU (road side unit) which contains all the information about traffic sign and different stop signals. Each vehicle is connected with RSU and as well as with the other vehicle which are within the communication range. In the system design of VANET there are some key factors [7] **Error! Reference source not found.** need to be considered those are as follows:

- Posted speed limit: The first factor is then posted speed limit. One needs to communicate any information if any vehicle is traveling above the posted speed limit. Drivers do not need be noticed if the road is clear. They can only communicate if the traffic is busy.
- Expected speed: It is possible to extend the idea of expected speed beyond the posted speed limit. Traffic congestion has very predictable trends which can be exploited. For example, major commuter routes will be slow during rush hour. If this information is available to all of the nodes of the network then each node only needs to communicate when the recorded speed is outside the variance of expected speed. This would reduce the communication significantly without any difference in information available to the end user.
- System adoption: This system does not expect that all drivers adopt it. In fact it can be designed to work with only a small fraction of total drivers participating in the network. While it might be desirable that all cars use VANET it is highly unlikely that will happen in the near future. It is far more acceptable that drivers from higher economic classes and professional drivers will be the first adopters.
- Serendipitous exchange: In the system each vehicle collects, stores and exchanges all traffic information that is made available to it. The nodes in the network do not differentiate between information that might be useful to the driver and information that will be of little use to the driver. By cooperating nodes will construct a more exact global model of the road network. While information held by one

vehicle of may be of no use to that car, it may serendipitously exchange that information to another car that values that information.

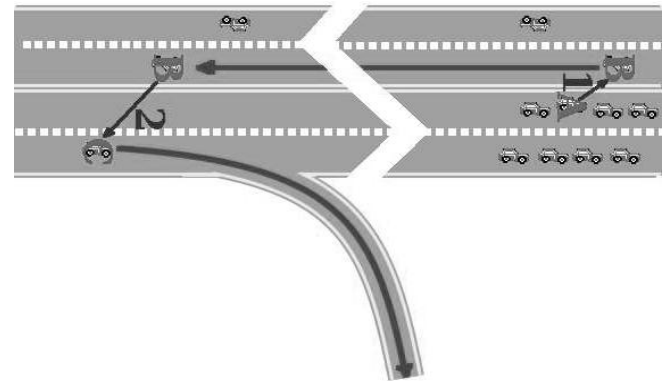


Fig. 6. An example of Serendipitous Exchange

An example for this system of information exchange is shown in fig. 6. Car A is trapped in traffic and drives a message to car B. That information is of little value to car B since the congestion is on the other side of the road. However a little downs the road B relays the information to car C. With that new information car C leaves the road for a better route.

V. CELLULAR AD-HOC NETWORKS

Cellular ad hoc networks are self-organizing multi-hop networks with multiple access points that are connected with a broadband core network. The difference of a cellular ad hoc network from an isolated ad hoc network is that most of the traffic in the cellular ad hoc network is to/from access points. After the standardization of the 3G (3rd generation) systems IMT-2000/UMTS, researchers are beginning to develop concepts and technologies for wireless beyond 3G systems. Higher transmission rate and lower system cost are regarded as the key requirements of the beyond 3G systems. A new air interface concept to form a W-CHAM13 network that is suited for the wireless multimedia communications beyond 3G is presented in [8]. Its main features are self-organizing, multi-hop transmission and QoS (Quality of Service) guarantee. As major applications of beyond 3G systems will require a transmission rate 10 times higher than that of 3G, the beyond 3G systems must use frequency bands over 3GHz in the view of availability and feasibility of frequency spectrum to support high rate services. The respective frequencies have very limited ability to penetrate obstructions and have very irregular propagation characteristics so that frequency planning is very difficult there. In addition, higher transmission rates and higher frequency bands result in a very limited communication range and irregular radio coverage. Two approaches might be used to achieve reasonable radio coverage. One is to increase the number of base stations and the output power. This method might increase the system cost significantly. The other one uses cellular ad hoc networks with multi-hop transmissions to extend the radio coverage and realize a cost-effective solution.

VI. OTHER APPLICATIONS

One of many possible uses of mobile ad-hoc networks is in some business environments, where the need for collaborative computing might be more important outside the office environment than inside, such as in a business meeting outside the office to brief clients on a given assignment. Wireless ad-hoc network is also very effective in class room where students and teachers can communicate with each other without involving the main network. Fig. 7 shows a wireless ad-hoc networking in a class room. In this environment students can download desired content, review lecture notes and begin their work on assignment.

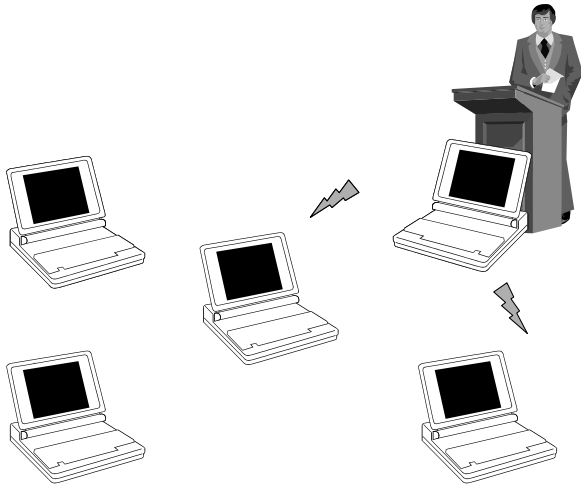


Fig. 7. Wireless ad-hoc networking in class room

A commercial application, such as Bluetooth, is one of the recent developments utilizing the concept of mobile ad-hoc networking. Bluetooth was first introduced in 1998. Bluetooth uses radio waves to transmit wireless data over short distances. It can support many users in any environment. Eight devices can communicate with each other in a small network known as Pico-net. At one time, ten of these Pico-nets can coexist in the same coverage range of the Bluetooth radio. A Bluetooth device can act both as a client and a server. A connection must be established to exchange data between any two Bluetooth devices. In order to establish a connection a device must request a connection with the other device.

Bluetooth was based on the idea of advancing wireless interactions with various electronic devices. Devices like mobile phones, personal digital assistants, and laptops with the right chips could all communicate wirelessly with each other. However, later it was realized that a lot more is possible. At present, Bluetooth technology is in used in a variety of different places. Not long ago, in May 2004, a service known as BEDD was launched in Singapore **Error! Reference source not found.** BEDD uses Bluetooth wireless communications to scan strangers' phones for their personal profiles. Once the software is downloaded into a compatible phone, it automatically starts searches for and exchanges profiles with other phones that come within a 20-meter radius.

Underwater Networking is also an important application of wireless ad-hoc networking. The main goal of underwater

ad-hoc networking is to allow divers/sensors to know where other divers/sensors, dive vessel are located, providing the ability for short communications between all the equipments, informing others if any emergency situation occurs. Fig. 8 shows an example of underwater ad-hoc networking.

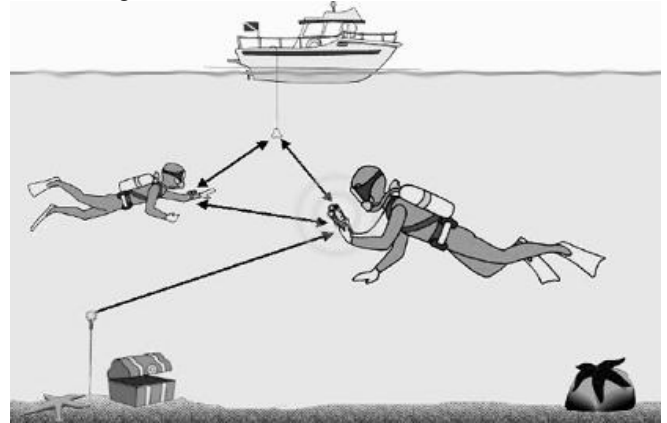


Fig. 8. Underwater Ad-hoc Networking

VII. CONCLUDING REMARKS

In conclusion, wireless ad-hoc networks allow the construction of flexible and adaptive networks with no fixed infrastructure. These networks are expected to play an important role in the future wireless generation. Future wireless technology will require highly-adaptive mobile networking technology to effectively manage multi-hop ad-hoc network clusters, which will not only operate autonomously but also will be able to attach at some point to the fixed networks. To meet the need of a member of ad-hoc group to access the global internet or to be reached over the global Internet, various ad-hoc gateways can be used to integrate ad-hoc networks with the global Internet. In cellular ad-hoc networks can improve the system performance as the bottleneck effect in the AP (Access Point) is reduced because of the reduced interference. The radio coverage of APs can be enlarged using multi-hop transmissions without the need of increasing transmission power so that the system cost can keep very low. The organization remains simple by limitation of the transmission hops. Frequency planning that is very difficult in the frequency range above 3G Hz can be avoided by self-organization of APs and mobile nodes. Cellular ad hoc networks seem to be a promising solution for broadband wireless access networks in beyond 3G systems.

The goal of this paper was to provide insight into the understanding of ad-hoc networking aspects and its applications based on different scenarios in a wireless environment. This was not geared toward any system in particular. No system specific protocols, parameters or standard were discussed in this work. Wireless systems engineers, in their research of wireless data communication, can use this paper as a tool for understanding the application of ad-hoc networking.

REFERENCES

- [1] P. Chenna Reddy and Dr. P. Chandrasekhar Reddy, "Performance Analysis of Ad-hoc Network Routing Protocols", *Academic Open Internet Journal*, ISSN 1311- 4360, Volume 17, 2006.
- [2] Tim Bouge, "Ad-hoc Networking in Military Scenarios". Thesis. May 2004.
- [3] <http://en.wikipedia.org/wiki/HIPAA> enacted by the US.Congress, 1996.
- [4] <http://mobility.tamu.edu/ums> Annual Urban Mobility Report. 2007.
- [5] <http://www.bedd.com/corporate.html> BEDD corporate report. May 2004.
- [6] Petteri Kuosmanen, "Classification of Ad Hoc Routing Protocols". Finnish Defence Forces Naval Academy, 2004.
- [7] Philippe Jacquet, Paul Muhlethaler, Amir Qayyum, Anis Laouiti, Laurent Viennot, Thomas Clausen. "Optimized Link State Routing Protocol." RFC 3626, October 2003.
- [8] B. Xu., B. Walke, "A New Air Interface Concept for Wireless Multimedia Communications beyond the 3rd Generation", *Wireless Personal Communications*, vol. 23, no. 1, Oct. 2002, pp [121-135]

Author Biography

Md. Taslim Arefin received his B.Sc. in Computer Engineering from American International University –Bangladesh (AIUB) in 2005. He obtained his M.Sc. in Electrical Engineering – Specialization Telecommunications from Blekinge Institute of Technology (BTH), Sweden in 2008. At the present time he is working as Senior Lecturer in the Department of Electronics & Telecommunication Engineering at Daffodil International University, Dhaka, Bangladesh.

Commenting the Virtual Memory Management Kernel Source Code 2.6.31 for Educational Purpos

Archana S. Sumant , Pramila M.Chawan

Abstract- While doing a study of Operating System one can easily learn theoretical concepts but if anyone wants to study an actual operating system code there is only one way that is the documentation which comes with an operating system source code. As you can easily download a Linux source code, you will found that very less document is available for doing a study of Virtual Memory manager .Practically if any one wants to study the techniques used in designing an Operating System then it will take a long time to understand the implementation. So what I have done in my project is that the detail documentation of kernel code from virtual memory management point of view is presented to those who wants to study an VM part of an operating system Linux kernel 2.6.31.

This project deals with the study of Virtual Memory Manager in Linux kernel 2.6.31 and some comparative conclusions between 2.4 and 2.6 kernel features.

Index Terms—page allocator, anonymous memory ,page cache ,memory allocators , Copy on Write (CoW).

I. INTRODUCTION

The memory management subsystem is one of the most important parts of the operating system. Since the early days of computing, there has been a need for more memory than exists physically in a system. Strategies have been developed to overcome this limitation and the most successful of these is virtual memory. Virtual memory makes the system appear to have more memory than it actually has by sharing it between competing processes as they need it. Virtual memory is implemented in Linux with secondary storage disks as extension so that the memory size can be increased according to program need though system have physical RAM size less. The kernel will write the

contents of a currently unused block of memory to the hard disk so that the memory can be used for another purpose. When the original contents are needed again, they are read back into memory. This is all made completely transparent to the user; programs running under Linux only see the larger amount of memory available and don't notice that parts of them reside on the disk from time to time.

Memory management is one of the most complex and at the same time most important parts of the kernel. It is characterized by the strong need for cooperation between the processor and the kernel because the tasks to be performed require them to collaborate very closely [9].

Figure 1 gives a conceptual overview on how basic Linux memory management works. Central to all memory management is the *page allocator*. The page allocator can hand out pieces of memory in chunks of *page size* bytes. The page size is fixed in hardware at 4 KBytes for i386, x64 and many other architectures. The page size is configurable on several platforms. On Itanium the page size is usually configured to be 16k. All other memory allocators are in one way or another based on the page allocator and take pages out of the pool of pages managed by the page allocator. The page allocator may provide pages that are mapped into a processes virtual address space. There are two ways that pages mapped into user space are used. The first type of pages is used for *anonymous memory*. These are pages for temporary use while a process is running. They are not associated with any file and are typically used for variables, the heap and the stack. Anonymous memory is light weight and can be managed in a more efficient way than file backed pages because no mappings to disk (which require serialization to access) have to be maintained.

Anonymous memory is private to a process (hence the name) and will be freed when a process terminates. Anonymous memory may be temporarily moved to disk (swapping) if memory becomes very tight. However, at that point an anonymous page acquires a reference to swap space and therefore a mapping to secondary

Archana S. Sumant is with the Veermata Jijabai Technological Institute ,Matunga , Mumbai 400019 (INDIA).
Phone: +91 9503666033
Email: archana.s.vaidya@gmail.com

Pramila M.Chawan, is with the Veermata Jijabai Technological Institute , Matunga,Mumbai 400019 (INDIA).
Phone: +91 9869074620
Email: pmchawan@vjti.org.in

storage, which adds overhead to the future processing of this page.

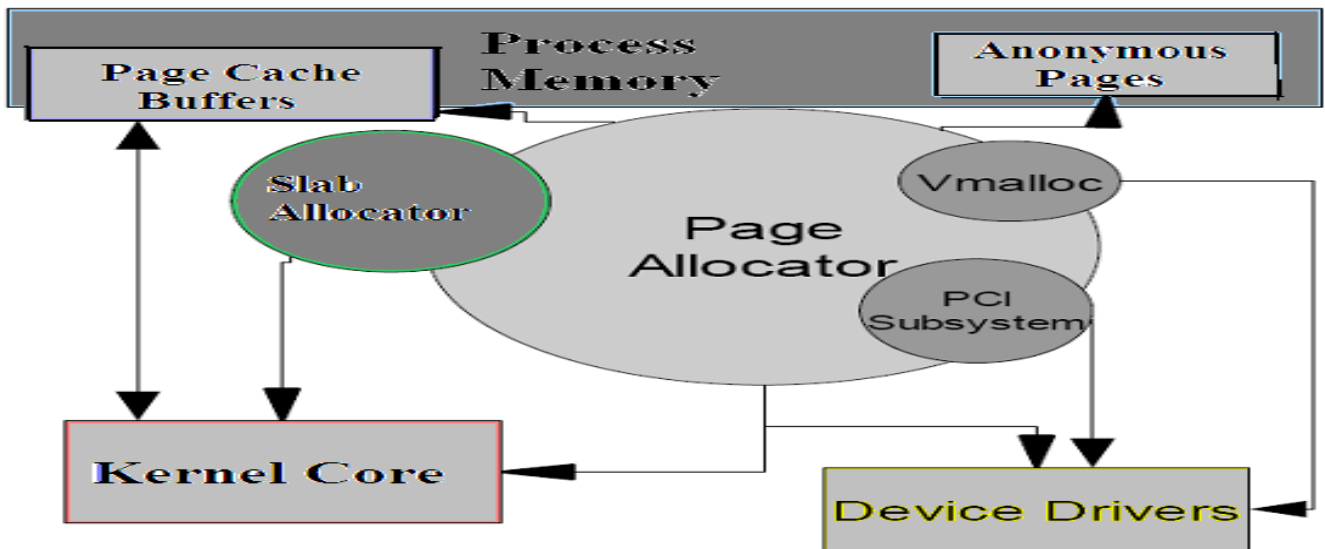


Figure 1 : Linux Memory Subsystems

The *page cache* or *buffers* are pages that have an associated page on a secondary storage medium such as a disk. Page cache pages can be removed if memory becomes tight because their content can be restored by reading the page from disk. Most important is that the page cache contains the executable code for a process. A process may map additional files into its address space via the `mmap()` system call. Both the executable code and the mapped files are directly addressable via virtual addresses from the user process. The operating system may also maintain buffers in the page cache that are not mapped into the address space of a process. This frequently occurs if a process manipulates files through system calls like `sys_write()`, `sys_read()` that read and modify the contents of files. The unmapped pages may be also thought as belonging in some loose form to a process. However, all page cache pages may be mapped or accessed by multiple processes and therefore the ownership of these pages cannot be clearly established.

The *kernel core* itself may need pages in order to store meta data. For example file systems may use buffers to track the location of sections of a file on disk, pages may be used to establish the virtual to physical address mappings (page tables) and so on. The kernel also needs to allocate memory for structures of varying sizes that are not in units of the page size in use on the system. For that purpose the *slab allocator* is used. The slab allocator retrieves individual pages or contiguous ranges of pages from the page allocator but then uses its own control structures to be able to hand out memory chunks of varying sizes as requested by the kernel or

drivers. The slab allocator employs a variety of caching techniques that result in high allocation performance for small objects. Slab allocations are used to build up structures that maintain the current system state. This includes information about open files, recently used filenames and a variety of other state objects.

The *device drivers* utilize both the page allocator and the slab allocator to allocate memory to manage devices. There are a couple of additional variations on page sized allocations for device drivers. First there is the *vmalloc* subsystem. Vmalloc allows the allocation of larger chunks of memory that appear to be virtually contiguous within kernel context but the actual pages constituting this allocation may not be physically contiguous. Therefore vmalloc can generate a virtually contiguous memory for large chunks of memory even if the page allocator cannot satisfy request for large contiguous chunks of memory anymore because memory has become fragmented. Accesses to memory obtained via the vmalloc allocator must use a page table to translate the virtual addresses to physical addresses and may be not as efficient as using a direct physical address as handed out from the page allocator. Vmalloc memory may not be mapped into user space. Finally, the PCI subsystem itself may can be used by a device driver to request memory that is suitable for DMA transfers for a given device via `dma_alloc_coherent()`. The way of obtaining that type of memory varies with the type of underlying hardware and therefore the allocation technique varies for each platform supported by Linux.

Table 1 Basic Memory Allocators under Linux

<i>Allocation Function</i>	<i>Allocator</i>	<i>Action</i>
<i>alloc_pages(flags, order)</i>	Page allocator	Allocates 2 ^{order} contiguous pages
<i>kmalloc(size, flags)</i>	Slab allocator	Allocate <i>size</i> bytes of memory
<i>kmem_cache_alloc(cache, flags)</i>	Slab allocator	Allocate an entry for the indicated slab
<i>vmalloc(size)</i>	vmalloc subsystem	Allocate a virtual contiguous memory area with a minimum of <i>size</i> bytes.
<i>dma_alloc_coherent(device, size, &addr, flags)</i>	PCI subsystem/ architecture specific support	Allocate DMA capable memory for the indicated device.

Table 1 gives an overview of the basic memory allocators under Linux:

II. COMMENTARY ON KERNEL CODE

To get a comprehensive view on how the kernel works, one is required to read through the source code line by line. This project focus on giving detail documentation of kernel code 2.6.31 so that the time to understand the kernel functions will be measured in weeks and not months. For managing such huge source code I have used a LXR tool which can be downloaded from <http://lxr.linux.no/>.

The code commentary will be done according following flow.

- 1 Boot Memory Allocator
 - 1.1 Representing the Boot Map
 - 1.2 Initializing the Boot Memory Allocator
 - 1.3 Allocating Memory
 - 1.4 Freeing Memory
- 2 Physical Page Management
 - 2.1 Allocating Pages
 - 2.2 Free Pages
 - 2.3 Page Allocate Helper Functions
 - 2.4 Page Free Helper Functions
- 3 Non-Contiguous Memory Allocation
 - 3.1 Allocating A Non-Contiguous Area
 - 3.2 Freeing A Non-Contiguous Area
- 4 Slab Allocator
 - 4.1 Introduction
 - 4.2 Slabs
 - 4.3 Objects
 - 4.4 Sizes Cache
 - 4.5 Per-CPU Object Cache
 - 4.6 Slab Allocator Initialization
 - 4.7 Interfacing with the Buddy Allocator

5 Process Address Space

- 5.1 Managing the Address Space
- 5.2 Process Memory Descriptors
 - 5.2.1 Allocating a Descriptor
 - 5.2.2 Initializing a Descriptor
 - 5.2.3 Destroying a Descriptor
- 5.3 Memory Regions .
 - 5.3.1 Creating A Memory Region
 - 5.3.2 Finding a Mapped Memory Region
 - 5.3.3 Finding a Free Memory Region
 - 5.3.4 Inserting a memory region
 - 5.3.5 Merging contiguous region
 - 5.3.6 Remapping and moving a memory region
 - 5.3.7 Locking a Memory Region
 - 5.3.8 Unlocking the region
 - 5.3.9 Fixing up regions after locking/unlocking
 - 5.3.10 Deleting a memory region
 - 5.3.11 Deleting all memory regions
- 5.4 Page Fault Handler
 - 5.4.1 Handling the Page Fault
 - 5.4.2 Demand Allocation
 - 5.4.3 Demand Paging
 - 5.4.4 Copy On Write (COW) Pages

III. APPLICATIONS

- 1.This will reduce the amount of time a developer or researcher needs to understand Linux Virtual memory manager.
- 2.Further this study and documentation can be used to improve some aspects of Virtual memory management.
- 3.Developer can even change particular part of Virtual memory manager code for particular applications .

IV. CONCLUSION

Very little help or code documentation available for practically understanding an operating system one may require an extra time for doing so. My project gives

detail study of kernel code 2.6.31 from point of view of virtual memory manager (architecture independent features) and will help to those who wants to swim in operating system code .In future enhancement one can modify the kernel code and recompile it to make new version.

Computer Architecture & Operating Systems. She has published 14 papers in National Conferences & 4 papers in International Conferences & Journals. She has guides 25 M. Tech. projects & 75 B. Tech. projects. Currently she is guiding Ms. Archana Sumant's M. Tech. project named "Virtual Memory Management in Linux kernel 2.6".

REFERENCES

- [1] <http://lwn.net/> Linux info from the source
- [2] Gorman Mel. "Understanding the Linux Virtual Memory Manager " Prentice Hall Professional Technical Reference 2004
- [3] <http://kernel.org/doc>
- [4] <http://www.linuxhq.com/kernel/v2.4/index.html>
- [5] <http://www.linuxhq.com/kernel/v2.6/index.html>
- [6] Neil Horman Understanding Virtual Memory In Red Hat Enterprise Linux 4 Version 0.1 –
- [7] <http://www.perens.com/Book> (Mel Gorman book site)
- [8] The Linux Kernel Source Tree. Version 2.6.31
<http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.31.5.tar.bz2>
- [9] Wolfgang Mauerer "Professional Linux® Kernel Architecture "Wiley Publishing,



Archana S. Sumant is currently doing her M.Tech at "Veermata Jijabai Technological Institute ,Matunga , Mumbai (INDIA) and received Bachelors' Degree in Computer science and Engineering from "Walchand College Of Engineering ",Sangli (INDIA) in 2002. Her areas of interest are Operating System and Database management System. She is life member of ISTE (Indian Society Of Technical Education).She has authored 4 National and One International papers in Conferences.



Pramila M.Chawan is currently working as an Assistant Professor in the Computer Technology Department of "Veermata Jijabai Technological Institute (V. J. T. I.), Matunga, Mumbai (INDIA)". She received her Masters' Degree in Computer Engineering from V. J. T. I., Mumbai University (INDIA) in 1997 & Bachelors' Degree in Computer Engineering from V. J. T. I., Mumbai University (INDIA) in 1991 .She has an academic experience of 18 years (since 1992). She has taught Computer related subjects at both the (undergraduate & post graduate) levels. Her areas of interest are Software Engineering,

Efficient Service Retrieval from Service Store using Map Reduce

K.V. Augustine, S.K.V Jayakumar,

Department of Computer Science
Pondicherry University
Puducherry
augustine.k.v@gmail.com

Abstract— The paradigm shift from proprietary standards to Global standards in service computing has made web a user centric environment. With the advent of Web service more and more user are on track to team up and share for their benefits. In a user centric environment the service user namely the consumer, an application developer in our context is oriented towards providing more value added services by means of discovery and composition. Here finding all similar services that matches user demand is desired.. Homogenous or similar services at fine grained level are clustered based on their functional similarity. These are stored in a service store called service aggregator which serves as a repository for the consumer to spot out his interest. Similar service discovery includes search of similar single operation or composite operation. In our approach we propose a storage model that can incorporated functional as well as QoS i.e. non functional characteristics so that more specific user demand can be contented. To reduces the time consumed in searching the store we propose map reduce based searching framework. It not only reduces the time consumed in search but also helps the user to match is demand more precisely.

Keywords – Service Discovery, Similar Service, Service Store, Relational model, QoS, MapReduce

1 Introduction

Service oriented computing has gained momentum with the paradigm shift from proprietary standard to global standard. The advent of web services has further boosted the environment into more user centric were more and more user collaborate for their benefits. The technology support that Web Service [1] provides using XML standards such UDDI, WSDL and SOAP has reduced the challenge in interoperability to data and communication level. This enabled the increase in number of services both in demand and offer. The competitive push of the providers and competence pull of consumers has made this environment an ever green research area.

As the number of services has increased the effort needed by the consumer to run through each service to find the service of his choice is taxing. To overcome this challenging an effective discovery mechanism is significant. The major challenge lies in digging out the optimal service precisely. The basic sources for discovery are UDDI where the description and location of the service is depicted and WSDL [4] where the interface, operations input and output parameters are provided. In traditional approach [5][12][15][17]the search is based on keyword in public UDDI registries, where the keywords in the user query are matched with the keywords in the description. This approach lack precision as it returns irrelevant services and

also misses relevant services. Moreover users are willing to be more precise in their request rather than keywords. This challenge has been addressed by providing semantic annotations to describe services using OWL-s [3] or WSDL semantic [14]. But again the ontology based approach suffers from performance problem due to ontology reasoners and WSDL approach suffers from precision problem. Moreover the dynamic nature of the services provider and service consumer has made service discovery an ever improving model.

In some context the user namely a valued added service developer may went through the process of service discovery and ends up in a service which he found inappropriate for some reasons, he may prefer to find similar operations that takes similar inputs/outputs to the ones just considered together with that best suites his preferences. It is reasonable to support such similar services search which assist in discovering similar service and user preferences. Such approach should be efficient for the consumers to find the desired service.

In our approach we would like to propose an efficient discovery methods that can complement with user centric environment where user demand matched to both functional and non functional characteristics of similar operations as well potentially composable operations. The approaches in the paper include

- Similarity measures based on the metadata available from the WSDL description and matching similar datatypes into concepts and algorithms to retrieve the underlying semantic. as in.[16][18]
- We propose a tuple based search model where searching of similar operations can be based on tuple matching using map reduce concepts. The approach can support both the retrieval of single similar operation and composable operations.

The remainder of this paper is organized as follows : Section 2 Literature survey 3 Proposed system 4 Extracting Similar operation section 5 Framework for service search Section 6 Experimental Evaluation Section 7 Conclusion and future direction.

2 Literature Survey

In discovery of similar services two type of services can be identified, single service that matches the user preference may be termed as atomic or elementary service or if a single service does not satisfies his request then set of service termed as composable services that satisfies his preferences.

2.1 Similarity measurement and similar concept clustering

The functionality offered by a service depends on its Operations. So the challenge lies in finding similar operations. The operations are similar if they have similar input and similar output. In case of sequence of operation where the intake of the first operation is the expected input and the outcome of the last operation is the desired output with intermediate operations using the output of its preceding operation. Due to heterogeneity nature in naming operation, input and output for instance “TravelByDestination” and “CabByLocation” both represent similar type of service with similar operation “BookTicket” and “ReserveTicket”. If input parameter to the first operation is “Location” and to that of the second operation is “Area” which are similar in nature. The challenge lies in

- Finding an appropriate similarity measure ie finding association between similar service, similar operation, similar input and similar output for example “book” and “reserve”.
- Clustering similar terms into concepts for example giving common concept name ticket {book, Reserve}.

The similarity measurement approach proposed in [20] is based on scheme matching of the input and output parameters; schema matching has less precision because it does not consider the underlying semantics. In [16] [18] similarity measurement is proposed based on association rules and clustering concepts using agglomeration algorithm. In both the approach the similarity is measured using the measures of support and confidence. In [18] for further improving clustering they employ domain taxonomy. The approach similar to [18] is proposed in our work as it has more precision compared to the other two approaches[20][16]. Multiple sources of evidence for concept clustering which can incorporate domain ontology and web contents will also considered.

2.2 Storage model for simple and composable services

As there exists two types of similar services namely single service or composable service two types of search has to be done. An efficient storage model that can incorporate both the type of searches is expected.

In [18] they proposed a directed graph based search model representing each operation as a node and the composition opportunity as directed edges, and assign the weight of the edge with similarity matching score between input and output but this approach lack scoring based on multiple attributes as the increase in number edges between the nodes is inversely proportional to the efficiency of the search. In [19] tuple based storage model is proposed but it does not include both the types of service ie single and composable. In our approach we would like to propose a relational model that can incorporate both the type of services and also the QoS preferences which is lacking in [18].

2.3 Quality of Service Modeling

QoS of web service are described using QoS description languages [7] which may be an ontology language like DAML QoS [2] or syntactical language like WSOL [13]. QoS model can be classified based on different aspects such as

performance, security, stability, user satisfaction [9]. Here in our approach we adopt the QoS aspects based on Performance and Stability quality along with cost and reputation

2.3.1 Performance Quality

Response Time: The time taken to send a request and to receive the response. The Response Time is measured at an actual Web service call and it can be calculated as difference between request completion time and user request time

$$P_{Resp} = R_{comp} - U_{reqs} \quad (1)$$

2.3.2 Stability quality

Successability: Successability is defined as the extent to which Web services yield successful results over request messages. It is the ratio of successfully returned messages after requested tasks are performed without errors.

$$S_{sucs} = \frac{M_{RESP}}{M_{REQS}} \quad (2)$$

2.3.3 Others factors

Execution Cost: it is termed as the amount of fee the provider charges for utility of his service it can be represented by Q_{cost}

Reputation: The value of the reputation is defined as the average ranking given to the service by consumers. It can be calculated as the ratio of average of user rank to total number of users

$$Q_{REPU} = \frac{R_{USER}}{N_{USER}} \quad (3)$$

The qualities described above give the quality measurement metrics from elementary services in case of composite services the quality should be based on the aggregation[3] of the above metrics which is depicted in table II.

3 Proposed System

3.1 Overall framework for service aggregator with definition of terms

In the section we describe the overall framework of our proposed system which is given in Fig1 and brief description of each component in it.

Keyword based crawler engine It is the search engine that extracts the service URL from the repositories based on some keyword matching.

WSDL extractor: Based on the URL’s the corresponding WSDL files are extracted from the service provider’s site.

I/O similarity matcher: The similarity matcher matches the input the similar based on some IR matching methods like TF/IDF can creates a bag of terms

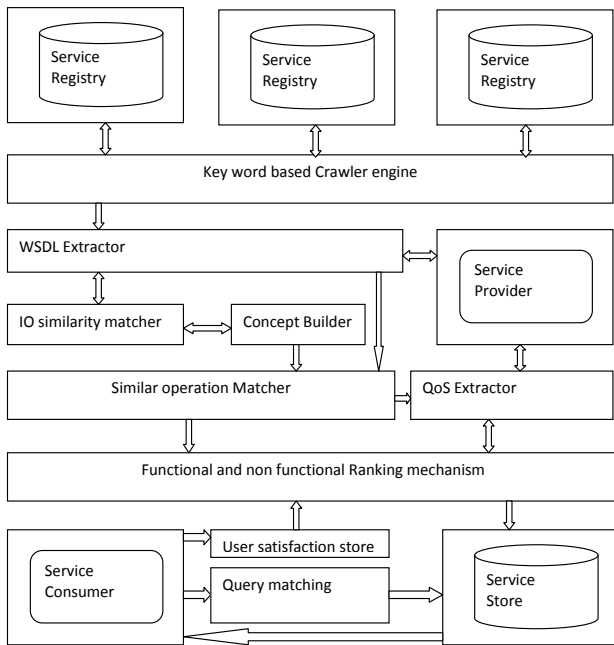


Fig 1: Overall framework for Service aggregator Model

Concept Builder: It is the work of the concept builder to find the association between the terms and clusters them into concepts.

Similar operation matcher : Based on the similarity of the input, output and description of the operation similar operations are clustered together using some clustering techniques.

QoS extractor: The QoS extractor finds the quality of service aspects as provided by the service providers for the service that are clustered

Functional and non functional based ranking : Based on the QoS and similarity measure the service are stored in the service store for retrieval

User satisfaction store: User satisfaction can be graded based on some scale and service store can further refined based on the satisfaction

Query matcher: It is the matcher the matches the user query to find the similar services that or of interest to the user based on the requirement and constrains

3.2 Storage Model of Service aggregator

In this section we propose relational based storage structure that can store both type of operation ie atomic or composable. Here we make a common sense assumption that the number of composable operation should to restricted to some maximum threshold as the increase in number of operation may increase the execution time.

3.3.1 Basic tuple structure

We propose three basic tuple structure namely Similarity Index, Masterpool , QualityStore to be store in the service aggregator storage. The structure of each is detailed below

3.3.2 Structure of SimilarIndex

The SimilarIndex tuple is given by the structure $\langle S_{mcode}, S_{code}, O_{code}, I_{para}, O_{para}, R_{first}, R_{last} \rangle$, Where S_{mcode} is the code given to similar operation in the MaterPool, S_{code} is the code of the service containing the operation , O_{code} in the operation code, I_{para} is the input parameter to the operation and O_{para} is the output parameter to the operation. R_{first} and R_{last} are the first and last record of S_{mcode} .

3.3.3 Structure of MasterPool

The MasterPool tuple is given by the structure $\langle S_{mcode}, S_{code}, O_{code}, S_{name}, O_{name}, I_{para}, O_{para}, O_{cons}, O_{seq} \rangle$, sorted by S_{mcode} Where S_{mcode} is the code given to similar operation in the mater pool, S_{code} is the code of the service containing the operation , O_{code} in the operation code, I_{para} is the input parameter to the operation, O_{para} is the output parameter to the operation, S_{name} is the service name, O_{name} is the operation name, O_{cons} is construct of the operation atomic or composite and O_{seq} sequence of operation in case of composable operation.

3.3.4 Structure of QualityStore

The QualityStore tuple is given by the structure $\langle O_{code}, Q_{xmax}, Q_{xmin} \rangle$, where O_{code} is the operation code, Q_{xmax} , Max value of the quality and Q_{xmin} is the minimum value of the quality.

Table 1. Structure of Similar Index

Smcode	Scode	Ocode	lpara	Opara
SM1	S1	O1	ZipCode	Temperature

Table 2. Structure of quality store

Ocode	QE _{max} ,	QE _{min}
O1	0.56	0.45

Table3. Structure of Master Pool

Smcode	Scode	Ocode	Sname	Oname	lpara	Opara	Ocons	Oseq
SM1	S1	O1	Weather forecast	GetTempByZipcode	ZipCode	Temperature	atomic	null
SM1	S2	O2	-	-	Areacode	Warmth	Sequence	O3O4
SM2	S3	O3	Area locator	GetAreabyareacode	Areacode	Areaname	atomic	null
SM3	S4	O4	Climatefind	Getwarmthbyarea	Areaname	Warmth	atomic	null

4 Extracting Similar Operations

In this section we provide the method for similar operation extraction. We adhere to the same approach provided in [18] as it has higher precision when compared to the approaches in [20] and [16]. We propose that the improvement in clustering can be enhanced by comparing with multiple source of evidence.

The approach is based on the heuristics that name of the operations, input/output parameters is often combined as a sequence of terms eg. *ZipCodeToTemperture*. Another commonsense heuristics is that the words trends it express the same semantics concepts if they often occur together[20].

4.1 Similarity measurement for Text Description

The textual description of service, operation or input/output is done using the tradition IR techniques of TF/IDF where is the term frequency and is defined as ratio of number of occurrence of the term in the document to the total number of terms in the document.

4.2 Similarity measurement of input/output parameters

The input/output parameters are grouped into terms called the term bag. The association between two terms are measured in terms of two probability measures support and confidence. Support is the number of input/output containing the terms t_i to the total number of input/output terms and The association rules are computed using the A-Prior algorithm

4.3 Clustering association to concepts

Based on the association rule the terms are clustered into concept based on the measure (C_{ij}, S_{ij}) using agglomeration algorithm which is the bottom up version of hierarchical clustering.

The algorithm works as follows, each term is initialized to a cluster. It sorts the association rules in descending order of their confidence and then by support. The terms are arranged in the form of square matrix M where each M_{ij} is the tuple (C_{ij}, S_{ij}) . The each step the algorithm selects the highest ranking rule above some threshold δ ie $C_{ij} > \delta, S_{ij} > \delta$. The two terms are combined into a cluster and the values are adjusted. The algorithm terminates when no more ranking $> \delta$ are available. Finally each cluster is grouped under a concept.

4.4 Fine tuning the clustering of Concepts

The clustering algorithm above is an unsupervised bottom up approach. The clusters may have low cohesion due to the fact that only association rules are considered. In [3] it is proposed that domain taxonomy can be used for further refining the clusters. In their approach matching score was calculated and terms are clustered taking this score together with association for clustering. The domain taxonomy is build using a floksonomy. Here we suggest that domain ontology and more web contents can also be added for building up the domain taxonomy

4.5 Discovering similar operations

Operation are defined by the tuple $O_{op} = \langle D_{op}, D_{in}, D_{ou} \rangle$. The similarity of operations O_{op1} and O_{op2} are measured by finding the similarity of each term in the tuple.

To measure the text description of the service S_{ds} and the operation D_{op} we using traditional TF/IDF measure as given in the previous section.

Next the similarity of the input and output are measured by considering the under lying semantics. First the similarity of the description of the input/output names are evaluate using TF/IDF then each term in the input/output are replaced by their corresponding concept and the concepts are compared using TF/IDF measure.

Finally the similarity Score between the two O_{op1} and O_{op2} Score (O_{op1}, O_{op2}) is calculated by multiplying each similarity score ie the similarity score of service description score (S_{ds1}, S_{ds2}) , similarity score of operation description score (D_{op1}, D_{op2}) , similarity score of inputs score (D_{in1}, D_{in2}) and similarity score of outputs score (D_{ou1}, D_{ou2}) by a weighing factor w_i respectively for each score and then summing them.

w_i are considered in such a way that the sum of w_i is unity. If the Score $(O_{op1}, O_{op2}) > \omega$ where ω is the given threshold then the operation O_{op1} & O_{op2} are considered similar. aggregate store

5 Framework for Service search using Map Reduce

In.in this section we provide a framework as given in fig 2 which we propose to use to search similar services for the user demand. in order to improve the efficiency the framework is designed using the concepts of mapreduce[6].

5.1 Mapping user query

Map part

When the user submits the query in the form $Q = \langle I_{use}, O_{use}, P_{ex}, P_{cost}, \dots \rangle$, The similarity index table is partitioned into τ parts and each part is assigned to the function

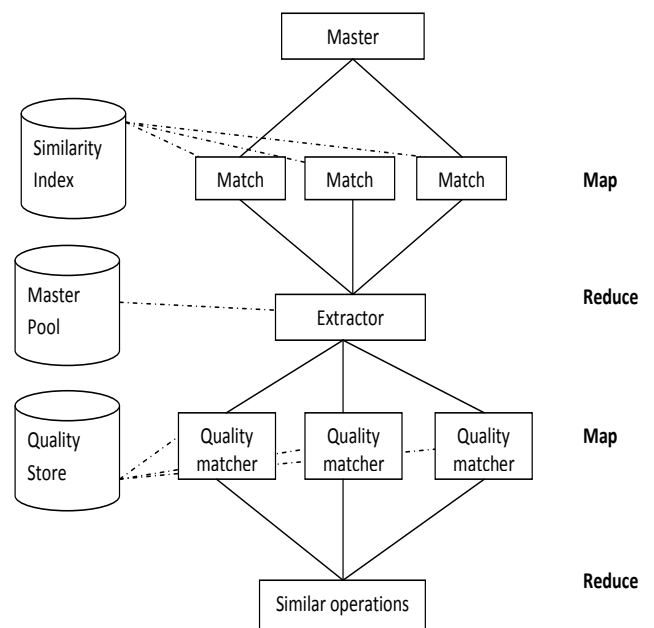


Fig 2: Framework for similar service search

Algorithm 1 :Match for matching user query and index table
 Input : $I_{use}, O_{use}, N_1, N_2, M_{sim}^i, SI_{code}^i$
 Output: $M_{sim}^i, SI_{code}^i, R_{first}, R_{last}$

1. *Initialize* $M_{sim}^i = 0, SI_{code}^i = null$
2. **FOR** j various from N_1 to N_2 and each O_{code}^j // from similarity index table
3. **COMPUTE** $score((I_{use}, O_{use}), (I_{para} O_{para}))$
4. **IF** $score((I_{use}, O_{use}), (I_{para} O_{para})) > M_{sim}^i$
5. **THEN** $M_{sim}^i = score((I_{use}, O_{use}), (I_{para} O_{para}))$
6. **ASSIGN** $SI_{code}^i = S_{mcode}^j$
7. **END IF**
8. **ENDFOR**
9. **Return** $M_{sim}^i, SI_{code}^i, R_{first}, R_{last}$

Fig 3: Matching user query with similarity Index.

$Match(I_{use}, O_{use}, N_1, N_2, M_{sim}^i, SI_{code}^i)$ Where $N_2 - N_1 = \tau$ and $i = 1, \dots, p$.

The value of τ is number of records in SimilarIndex divided by number of partitions p . N_1 and N_2 are first and last record of a particular partition. The output of the function will be the maximum similarity score M_{sim}^i , and the similarity code SI_{code}^i for that score. The algorithm is given in Fig 3.

Reduce part

In reduce part the output (M_{sim}^i, SI_{code}^i) for $i = 1$ to p returned by the Match function is given as an input to $Extractor(M_{sim}^i, SI_{code}^i, O_{code}^i)$ which returns the set of operation code O_{code}^i (where $i = 1$ to n) that has maximum SI_{code} value. The algorithm is given in Fig 4.

5.2 Retrieving similar operations

Map part

The operation code return is combined with each user preference and the send to the function $Matchquality(O_{code}^i, P_i)$

Algorithm 2 :Extractor the extract the operation that has maximum similarity score
 Input : $M_{sim}^i, SI_{code}^i, p$ (number of partitions)
 Output: set of OR_{code}

1. *Initialize* $O_{code}^j = null$
2. **FOR** i various from 1 to p
3. **ASSIGN** $SI_{code} = Max(M_{sim}^i, SI_{code}^i)$ // SI_{code}^i that has maximum M_{sim}^i value
4. **END FOR**
5. **IF** $S_{mcode} =, SI_{code}$
6. **FOR** j varies from R_{first} to , R_{last} // records for Masterpool
7. $OR_{code} = OR_{code} \cup O_{code}$
8. **END FOR**
9. **ENDIF**
10. **RETURN** OR_{code}

Fig 4: Extracting operation from master pool

where i varies from 1 to n and $j=1$ to m where m is the number of preference. The number of partitions depends on the value of j . The algorithm is given in Fig 5.

The output of the matchquality function is the set of operation the satisfies the user preference for each preference

Reduce part

The set returned by each of the match quality function is sent as the input of the similaroperation. The similaroperation function returns the intersection of all the returned operation by match quality function which is the set of similar operations the matches the user query.

6 Experimental Evaluation

We have implemented our prototype, named as ServicePool, to discover the homogeneous services based on the approach described above. we employ a server that manages the pool in aspects of service clustering, discovery and execution. The server records the services. registered in the pool, including the service URL, service names, operations and inputs/outputs. We present the user interface for service discovery and subscription, as shown in Figure 6. Such UI is so friendly that: (1) the users can do basic operation easily and the discovered results are presented to the user with whole QoS spectrum according to the users' preference. As Operation Name, Input, Output, Description of the operation, URL of the service and the quality details Given the users' request, ServicePool can return the operations with similar inputs and outputs,. Note that the returned results by ServicePool are evaluated by using the usual metrics employed in traditional IR communities, we evaluate our approach with the recall (r) and precision (p).Also we measure the retrival efficiency of the system from other system as we use simple relational model and Map reduce concept.

Algorithm 3 :Match quality for matching the operations with user preferences
 Input : OR_{code}, P_i
 Output: OP_{code}^i

1. *Initialize* $OP_{code} = null$
2. **FOR** each element in OR_{code}
3. **COMPARE** the value QE_{max}, QE_{min} in quality store with user preference P_i of quality i
4. **IF** match
5. **THEN** $Op_{code}^i = Op_{code} \cup O_{code}$
6. **END IF**
7. **ENDFOR**
8. **Return** Op_{code}^i // ordered by Op_{code}

Fig 5: Matching quality from quality table

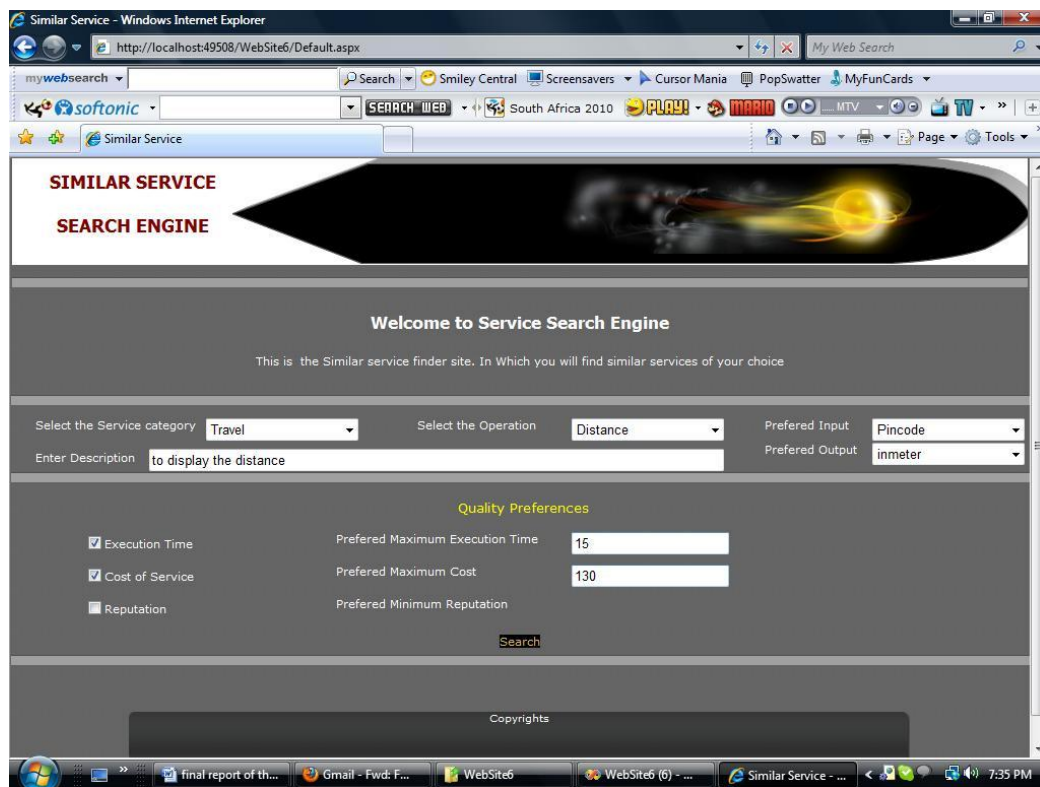


Fig 6: User interface for Service Search from Service Pool

Output of the System is

W1 : **Web Service:** Convertor
Description :Service that coverts various measurement values.
Operation: Currency Convertor
Descriptipon: The converts ruppees to dollars
Input: Rupees
Output: Dollars
Execution Time: 0.23 , **Cost:**\$75
 URL " http://WWW.convertor.com

W2 : **Web Service:** Convertor
Description :Service that coverts various measurement values.
Operation: Currency Convertor
Descriptipon: The converts Euro to dollars
Input: Euro
Output: Dollars
Execution Time: 0.23 , **Cost:**\$75
 URL " http://WWW.convertordol.com

7 Conclusion and future Direction

The advent of non proprietary standard, more precisely the proliferation of webservice has made web more usercentric more than a platform for business integration. So service computing has gained momentum and user started using the web more directly for their own benefits. It is evident that orientation towards finding similar services is significant. Here in our approach we have proposed a intermediate

storage area called service aggregator which can assist a user, here an application developer to find similar services to the service he have in hand. The WSDL metadata does not provide the semantic of the services it is essential to provide a similarity measurement approach the measure the similarity between different operation based on their description input/output parameters. The similarity measure provided [X.Liu et al. 2009] provides a better precision and recall we adhere to the same approach with little suggestion to improve the domain taxonomy. As for as storage model is concerned we have proposed a relational model which consists of three storage structure similarity index, master pool and quality store. To improve the efficiency we proposed MapReduce based matching and retrieving techniques. In our approach we have considered for single input and single output, the same can be extended for multiple input and multiple output in the future. In our approach we are considering only the WSDL description with the increase in semantic description of the web service using ontology language like OWL_S the similarity measure can be better enhanced resulting in better recall and precision.

REFERENCES

- [1]. BV Kumar, S.V. Subrahmanya "Web Service An introduction" Tata Mc-Graw-Hill Publishing Company Limited, New Delhi

- [2]. C. Zhou, L.-T. Chia, and B.-S. Lee, "DAML-QoS Ontology for WebServices," Proc. Int'l Conf. Web Services (ICWS '04), pp. 472-479, 2004.
- [3]. D. Martin, "OWL-S: Semantic Markup for Web Services," in Releases of DAML-S / OWL-S, 2004.
- [4]. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," 2001.
- [5]. Esynaps, <http://www.esynaps.com>, 2009
- [6]. J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In OSDI'04, 6th Symposium on Operating Systems Design and Implementation, Sponsored by USENIX, in cooperation with ACM SIGOPS, pages 137-150, 2004.
- [7]. K. Kritikos and D. Plexousakis "Mixed interger programming for QoS based web service matchmaking" IEEE transaction on service computing, Vol.2 No.2 April-June.
- [8]. Liangzhab Zeng, Boualem Benatallah, "QoS Aware Middleware for Web Service composition" IEEE transition of softwre Engineering VOL.30 No5 May 2004
- [9]. Quality Model for Web Services v2.0 Committee Draft, September 2005
- [10]. S. Robertson, "Understanding Inverse Document Frequency: On Theoretical Arguments for IDF," J. Documentation, vol. 60, no. 5, pp. 503-520, 2004.
- [11]. Selected Resources on Quality of Service (QoS) Specification and Contract-Based Management for XML Web Services 2006
- [12]. Strikeiron <http://www.strikeiron.com>, 2008.
- [13]. V. Tasic, B. Pagurek, and K. Patel, "WSOL—A Language for the Formal Specification of Classes of Service for Web Services," Proc. Int'l Conf. Web Services (ICWS '03), pp. 375-381, 2003.
- [14]. "Web Service Semantics," <http://www.w3.org/Submission/WSDL-S/>.
- [15]. WebServiceX, <http://www.webservicex.net>, 2009.
- [16]. Xing Dong, Alon Halvy, "Similarity search for web service" Porceeding of the 30th VLDB conference, Totronto Canda 2004
- [17]. Xmethods <http://www.xmethods.com>, 2009.
- [18]. Xuanzhe Liu, Gang Huang, "Discovering homogenous web service community in the user centric web environment", IEEE transition of service computing VOL.2 No2 April June 2009
- [19]. Xuanzhe Liu, Li Zhou, Gang Huang, Hong Mei "Consumer-Centric Web Services Discovery and Subscription" IEEE International Conference on e-Business Engineering
- [20]. Yanan Hao, Yanchim Zhang, "Web service discovery based on schema matching" ACSC conferences Australia 2007

Handwritten Character Recognition Using Bayesian Decision Theory

Vijiyakumar, Suresh Joseph

Department of computer science, Pondicherry University
Pondicherry-605014, India.
vijiya.kumar@gmail.com, sureshjosephk@yahoo.co.in

Abstract: Character recognition (CR) can solve more complex problem in handwritten character and make recognition easier. Handwriting character recognition (HCR) has received extensive attention in academic and production fields. The recognition system can be either online or offline. Offline handwritten character recognition is the sub fields of optical character recognition (OCR). The offline handwritten character recognition stages are preprocessing, segmentation, feature extraction and recognition. Our aim is to improve missing character rate of an offline character recognition using Bayesian decision theory.

Keywords: Character recognition, Optical character recognition, Off-line Handwriting, Segmentation, Feature extraction, Bayesian decision theory.

1. Introduction

The recognition system can be either on-line or off-line. On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitized or PDA, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwriting. Off-line handwriting recognition involves the automatic conversion of text in an image into letter codes which are usable within computer and text-processing applications. The data obtained by this form is regarded as a static representation of handwriting.

The aim of character recognition is to translate human readable character to machine readable character. Optical character recognition is a process of translation of human readable character to machine readable character in optically scanned and digitized text. Handwritten character recognition (HCR) has received extensive attention in academic and production fields.

Bayesian decision theory is a fundamental statistical approach that quantifies the tradeoffs between various decisions using probabilities and costs that accompany such decision.

They divided the decision process into the following five steps:

1. Identification of the problem.
2. Obtaining necessary information.
3. Production of possible solution.
4. Evaluation of such solution.
5. Selection of a strategy for performance.

They also include a sixth stage implementation of the decision. In the existing approach missing data cannot be

recognition which is useful in recognition historical data. In our approach we are recognition the missing words using Bayesian classifier. It first classifier the missing words to obtain minimize error. It can recover as much error as possible.

2. Related Work

The history of CR can be traced as early as 1900, when the Russian scientist Turing attempted to develop an aid for the visually handicapped [1]. The first character recognizers appeared in the middle of the 1940s with the development of digital computers. The early work on the automatic recognition of characters has been concentrated either upon machine-printed text or upon a small set of well-distinguished handwritten text or symbols. Machine-printed CR systems in this period generally used template matching in which an image is compared to a library of images. For handwritten text, low-level image processing techniques have been used on the binary image to extract feature vectors, which are then fed to statistical classifiers. Successful, but constrained algorithms have been implemented mostly for Latin characters and numerals. However, some studies on Japanese, Chinese, Hebrew, Indian, Cyrillic, Greek, and Arabic characters and numerals in both machine-printed and handwritten cases were also initiated [2].

The commercial character recognizers were available in the 1950s, when electronic tablets capturing the x-y coordinate data of pen-tip movement was first introduced. This innovation enabled the researchers to work on the on-line handwriting recognition problem. A good source of references for on-line recognition until 1980 can be found in [3].

Studies up until 1980 suffered from the lack of powerful computer hardware and data acquisition devices. With the explosion of information technology, the previously developed methodologies found a very fertile environment for rapid growth addition to the statistical methods. The CR research was focused basically on the shape recognition techniques without using any semantic information. This led to an upper limit in the recognition rate, which was not sufficient in many practical applications. Historical review of CR research and development during this period can be found in [4] and [3] for off-line and on-line cases, respectively.

The real progress on CR systems is achieved during this period, using the new development tools and methodologies,

which are empowered by the continuously growing information technologies.

In the early 1990s, image processing and pattern recognition techniques were efficiently combined with artificial intelligence (AI) methodologies. Researchers developed complex CR algorithms, which receive high-resolution input data and require extensive number crunching in the implementation phase. Nowadays, in addition to the more powerful computers and more accurate electronic equipments such as scanners, cameras, and electronic tablets, we have efficient, modern use of methodologies such as neural networks (NNs), hidden Markov models (HMMs), fuzzy set reasoning, and natural language processing. The recent systems for the machine-printed off-line [2] [5] and limited vocabulary, user-dependent on-line handwritten characters [2] [12] are quite satisfactory for restricted applications. However, there is still a long way to go in order to reach the ultimate goal of machine simulation of fluent human reading, especially for unconstrained on-line and off-line handwriting.

Bayesian decision Theory (BDT), one of the statistical techniques for pattern classification, to identify each of the large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within 20 fonts was randomly distorted to produce a file of 20,000 unique instances [6].

3. Existing System

In this overview, character recognition (CR) is used as an umbrella term, which covers all types of machine recognition of characters in various application domains. The overview serves as an update for the state-of-the-art in the CR field, emphasizing the methodologies required for the increasing needs in newly emerging areas, such as development of electronic libraries, multimedia databases, and systems which require handwriting data entry. The study investigates the direction of the CR research, analyzing the limitations of methodologies for the systems, which can be classified based upon two major criteria: 1) the data acquisition process (on-line or off-line) and 2) the text type (machine-printed or handwritten). No matter in which class the problem belongs, in general, there are five major stages Figure1 in the CR problem:

- 1) Preprocessing
- 2) Segmentation
- 3) Feature Extraction
- 4) Recognition
- 5) Post processing

3.1. Preprocessing

The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Preprocessing aims to produce data that are easy for the CR systems to operate accurately.

The main objectives of preprocessing are:

- 1) Noise reduction
- 2) Normalization of the data

3) Compression in the amount of information to be retained. In order to achieve the above objectives, the following techniques are used in the preprocessing stage.

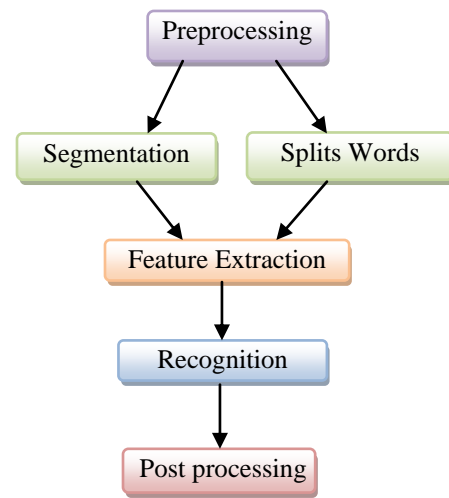


Figure 1. Character recognition

3.1.1 Noise Reduction

The noise, introduced by the optical scanning device or the writing instrument, causes disconnected line segments, bumps and gaps in lines, filled loops, etc. The distortion, including local variations, rounding of corners, dilation, and erosion, is also a problem. Prior to the CR, it is necessary to eliminate these imperfections. Hundreds of available noise reduction techniques can be categorized in three major groups [7] [8]:

- a) Filtering
- b) Morphological Operations
- c) Noise Modeling

3.1.2 Normalization

Normalization methods aim to remove the variations of the writing and obtain standardized data. The following are the basic methods for normalization [4] [10][16].

- a) Skew Normalization and Baseline Extraction
- b) Slant Normalization
- c) Size Normalization

3.1.3 Compression

It is well known that classical image compression techniques transform the image from the space domain to domains, which are not suitable for recognition. Compression for CR requires space domain techniques for preserving the shape information.

a) Threshold: In order to reduce storage requirements and to increase processing speed, it is often desirable to represent gray-scale or color images as binary images by picking a threshold value. Two categories of threshold exist: *global* and *local*. Global threshold picks one threshold value for the entire document image which is often based on an estimation of the background level from the intensity histogram of the image. Local (adaptive) threshold use different values for each pixel according to the local area information.

b) Thinning: While it provides a tremendous reduction in data size, thinning extracts the shape information of the characters. Thinning can be considered as conversion of off-line handwriting to almost on-line like data, with spurious branches and artifacts. Two basic approaches for thinning are 1) *pixel wise* and 2) *nonpareil wise* thinning [1]. Pixel wise thinning methods locally and iteratively process the image until one pixel wide skeleton remains. They are very sensitive to noise and may deform the shape of the character. On the other hand, the no pixel wise methods use some global information about the character during the thinning. They produce a certain median or centerline of the pattern directly without examining all the individual pixels. In clustering-based thinning method defines the skeleton of character as the cluster centers. Some thinning algorithms identify the singular points of the characters, such as end points, cross points, and loops. These points are the source of problems. In a nonpareil wise thinning, they are handled with global approaches. A survey of pixel wise and nonpareil wise thinning approaches is available in [9].

3.2. Segmentation

The preprocessing stage yields a “clean” document in the sense that a sufficient amount of shape information, high compression, and low noise on a normalized image is obtained. The next stage is segmenting the document into its subcomponents. Segmentation is an important stage because the extent one can reach in separation of words, lines, or characters directly affects the recognition rate of the script. There are two types of segmentation: external segmentation, which is the isolation of various writing units, such as paragraphs, sentences, or words, and internal segmentation, which is the isolation of letters, especially in cursively written words.

1) *External Segmentation*: It is the most critical part of the document analysis, which is a necessary step prior to the off-line CR. Although document analysis is a relatively different research area with its own methodologies and techniques, segmenting the document image into text and non text regions is an integral part of the OCR software. Therefore, one who works in the CR field should have a general overview for document analysis techniques. Page layout analysis is accomplished in two stages: The first stage is the *structural analysis*, which is concerned with the segmentation of the image into blocks of document components (paragraph, row, word, etc.), and the second one is the *functional analysis*, which uses location, size, and various layout rules to label the functional content of document components (title, abstract, etc.) [12].

2) *Internal Segmentation*: Although the methods have developed remarkably in the last decade and a variety of techniques have emerged, segmentation of cursive script into letters is still an unsolved problem. Character segmentation strategies are divided into three categories [13] is Explicit Segmentation, Implicit Segmentation and Mixed Strategies.

3.3. Feature Extraction

Image representation plays one of the most important roles in a recognition system. In the simplest case, gray-level or binary images are fed to a recognizer. However, in most of the recognition systems, in order to avoid extra complexity and to increase the accuracy of the algorithms, a more

compact and characteristic representation is required. For this purpose, a set of features is extracted for each class that helps distinguish it from other classes while remaining invariant to characteristic differences within the class [14]. A good survey on feature extraction methods for CR can be found [15]. In the following, hundreds of document image representations methods are categorized into three major groups are Global Transformation and Series Expansion, Statistical Representation and Geometrical and Topological Representation.

3.4. Recognition Techniques

CR systems extensively use the methodologies of pattern recognition, which assigns an unknown sample into a predefined class. Numerous techniques for CR can be investigated in four general approaches of pattern recognition, as suggested in [16] are Template matching, Statistical techniques, and Structural techniques and Neural networks.

3.5. Post Processing

Until this point, no semantic information is considered during the stages of CR. It is well known that humans read by context up to 60% for careless handwriting. While preprocessing tries to “clean” the document in a certain sense, it may remove important information, since the context information is not available at this stage. The lack of context information during the segmentation stage may cause even more severe and irreversible errors since it yields meaningless segmentation boundaries. It is clear that if the semantic information were available to a certain extent, it would contribute a lot to the accuracy of the CR stages. On the other hand, the entire CR problem is for determining the context of the document image. Therefore, utilization of the context information in the CR problem creates a chicken and egg problem. The review of the recent CR research indicates minor improvements when only shape recognition of the character is considered. Therefore, the incorporation of context and shape information in all the stages of CR systems is necessary for meaningful improvements in recognition rates.

4. The proposed System Architecture

The proposed research methodology for off-line cursive handwritten characters is described in this section as shown in Figure 2.

4.1 Preprocessing

There exist a whole lot of tasks to complete before the actual character recognition operation is commenced. These preceding tasks make certain the scanned document is in a suitable form so as to ensure the input for the subsequent recognition operation is intact. The process of refining the scanned input image includes several steps that include: Binarization, for transforming gray-scale images in to black & white images, scraping noises, Skew Correction-performed to align the input with the coordinate system of the scanner and etc., The preprocessing stage comprise three steps:

- (1) Binarization
- (2) Noise Removal

(3) Skew Correction

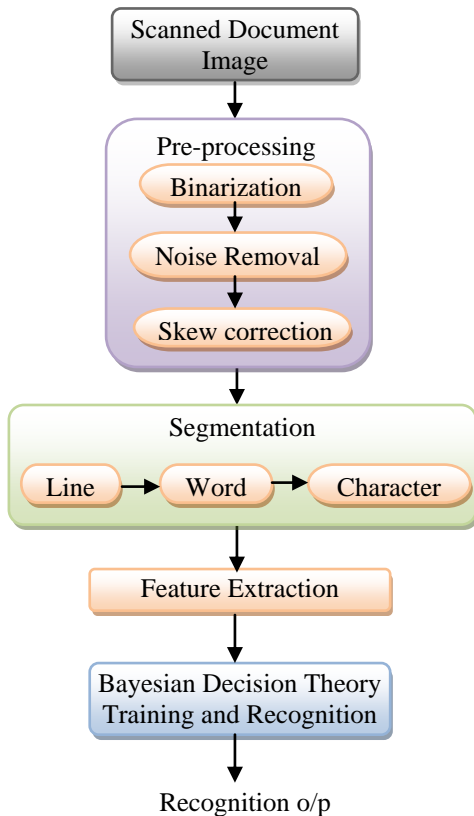


Figure 2. Proposed System Architecture

4.1.1 Binarization

Extraction of foreground (ink) from the background (paper) is called as threshold. Typically two peaks comprise the histogram gray-scale values of a document image: a high peak analogous to the white background and a smaller peak corresponding to the foreground. Fixing the threshold value is determining the one optimal value between the peaks of gray-scale values [1]. Each value of the threshold is tried and the one that maximizes the criterion is chosen from the two classes regarded as the foreground and back ground points.

4.1.2 Noise Removal

The presence of noise can cost the efficiency of the character recognition system; this topic has been dealt extensively in document analysis for typed or machine-printed documents. Noise may be due the poor quality of the document or that accumulated whilst scanning, but whatever is the cause of its presence it should be removed before further Processing. We have used median filtering and Wiener filtering for the removal of the noise from the image.

4.1.3 Skew Correction

Aligning the paper document with the co-ordinate system of the scanner is essential and called as skew correction. There exist a myriad of approaches for skew correction covering correlation, projection, profiles, Hough transform and etc.

For skew angle detection Cumulative Scalar Products (CSP) of windows of text blocks with the Gabor filters at different orientations are calculated. Alignment of the text line is used as an

important feature in estimating the skew angle. We calculate CSP for all possible 50X50 windows on the scanned document image and the median of all the angles obtained gives the skew angle.

4.2 Segmentation

Segmentation is a process of distinguishing lines, words, and even characters of a hand written or machine-printed document, a crucial step as it extracts the meaningful regions for analysis. There exist many sophisticated approaches for segmenting the region of interest. Straight-forward, may be the task of segmenting the lines of text in to words and characters for a machine printed documents in contrast to that of handwritten document, which is quiet difficult. Examining the horizontal histogram profile at a smaller range of skew angles can accomplish it. The details of line, word and character segmentation are discussed as follows.

4.2.1 Line Segmentation

Obviously the ascenders and descenders frequently intersect up and down of the adjacent lines, while the lines of text might itself flutter up and down. Each word of the line resides on the imaginary line that people use to assume while writing and a method has been formulated based on this notion shown fig.3.

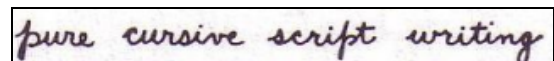


Figure 3. Line Segmentation

The local minima points are calibrated from each Component to approximate this imaginary baseline. To calculate and categorize the minima of all components and to recognize different handwritten lines clustering techniques are deployed.

4.2.2 Word and Character Segmentation

The process of word segmentation succeeds the line separation task. Most of the word segmentation issues usually concentrate on discerning the gaps between the characters to distinguish the words from one another other. This process of discriminating words emerged from the notion that the spaces between words are usually larger than the spaces between the characters in fig 4.

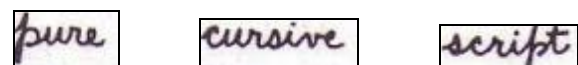


Figure 4. Word Segmentation

There are not many approaches to word segmentation issues dealt in the literature. In spite of all these perceived conceptions, exemptions are quiet common due to flourishes in writing styles with leading and trailing ligatures. Alternative methods not depending on the one-dimensional distance between components, incorporates cues that humans use. Meticulous examination of the variation of spacing between the adjacent characters as a function of the corresponding characters themselves helps reveal the writing style of the author, in terms of spacing. The segmentation scheme comprises the notion of expecting greater spaces between characters with leading and trailing ligatures.

Recognizing the words themselves in textual lines can itself help lead to isolation of words. Segmentation of words in to its constituent characters is touted by most recognition methods. Features like ligatures and concavity are used for determining the segmentation points.

4.3 Feature Extraction

The size inevitably limited in practice, it becomes essential to exploit optimal usage of the information stored in the available database for feature extraction. Thanks to the sequence of straight lines, instead of a set of pixels, it is attractive to represent character images in handwritten character recognition. Whilst holding discriminated information to feed the classifier, considerable reduction on the amount of data is achieved through vector representation that stores only two pairs of ordinates replacing information of several pixels. Vectorization process is performed only on basis of bi-dimensional image of a character in off-line character recognition, as the dynamic level of writing is not available. Reducing the thickness of drawing to a single pixel requires thinning of character images first. Character before and after Thinning After streamlining the character to its skeleton, entrusting on an oriented search process of pixels and on a criterion of quality of representation goes on the vectorization process. The oriented search process principally works by searching for new pixels, initially in the same direction and on the current line segment subsequently. The search direction will deviate progressively from the present one when no pixels are traced. The dynamic level of writing is retrieved of course with moderate level of accuracy, and that is object of oriented search. Starting the scanning process from top to bottom and from left to right, the starting point of the first line segment, the first pixel is identified. According to the oriented search principle, specified is the next pixel that is likely to be incorporated in the segment. Horizontal is the default direction of the segment considered for oriented search. Either if the distortion of representation exceeds a critical threshold or if the given number of pixels has been associated with the segment, the conclusion of line segment occurs. Computing the average distance between the line segment and the pixels associated with it will yield the distortion of representation. The sequence of straight lines being represented through ordinates of its two extremities character image representation is streamlined finally. All the ordinates are regularized in accordance to the initial width and height of character image to resolve scale Variance.

4.4 Bayesian Decision Theories

The Bayesian decision theory is a system that minimizes the classification error. This theory plays a role of a prior. This is when there is priority information about something that we would like to classify.

It is a fundamental statistical approach that quantifies the tradeoffs between various decisions using probabilities and costs that accompany such decisions. First, we will assume that all probabilities are known. Then, we will study the cases where the probabilistic structure is not completely known. Suppose we know $P(w_j)$ and $p(x|w_j)$ for $j = 1, 2, \dots, n$. and measure the lightness of a fish as the value x .

Define $P(w_j|x)$ as the a posteriori probability (probability of the state of nature being w_j given the measurement of feature value x).

We can use the Bayes formula to convert the prior probability to the posterior probability

$$P(w_j|x) = \frac{p(x|w_j)p(w_j)}{p(x)}$$

$$\text{Where } p(x) = \sum_{j=1}^c p(x|w_j) p(w_j)$$

$P(x|w_j)$ is called the likelihood and $p(x)$ is called the evidence.

Probability of error for this decision

$$P(\text{error}|x) = \begin{cases} P(w_1|x) & \text{if we decide } w_2 \\ P(w_2|x) & \text{if we decide } w_1 \end{cases}$$

Average probability of error

$$P(\text{error}) = \int_{-\infty}^{\infty} p(\text{error}, x) dx$$

$$P(\text{error}) = \int_{-\infty}^{\infty} p(\text{error}, x) p(x) dx$$

Bayes decision rule minimizes this error because

$$P(\text{error}|x) = \min \{P(w_1|x), P(w_2|x)\}$$

Let $\{w_1, \dots, w_c\}$ be the finite set of c states of nature (classes, categories). Let $\{\alpha_1, \dots, \alpha_a\}$ be the finite set of a possible actions. Let $\lambda(\alpha_i|w_j)$ be the loss incurred for taking action α_i when the state of nature is w_j . Let x be the

D -component vector-valued random variable called the feature vector.

$P(x|w_j)$ is the class-conditional probability density function. $P(w_j)$ is the prior probability that nature is in state w_j . The posterior probability can be computed as

$$P(w_j|x) = \frac{p(x|w_j)p(w_j)}{p(x)}$$

$$\text{Where } p(x) = \sum_{j=1}^c p(x|w_j) p(w_j)$$

Suppose we observe x and take action α_i . If the true state of nature is w_j , we incur the loss $\lambda(\alpha_i|w_j)$.

The expected loss with taking action α_i is

$$R(\alpha_i|x) = \sum_{j=1}^c \lambda(\alpha_i|w_j) p(w_j|x) \text{ which is also called}$$

the conditional risk.

The general decision rule $\alpha(x)$ tells us which action to take for observation x . We want to find the decision rule that minimizes the overall risk

$$R = \int R(\alpha(x)|x) p(x) dx$$

Bayes decision rule minimizes the overall risk by selecting the action α_i for which $R(\alpha_i|x)$ is minimum. The resulting minimum overall risk is called the Bayes risk and is the best performance that can be achieved.

4.5 Simulations

This section describes the implementation of the mapping and generation model. It is implemented using GUI (Graphical User Interface) components of the Java programming under Eclipse Tool and Database storing data in Microsoft Access.

For given Handwritten image character and convert to Binarization, Noise Remove and Segmentation as shown in Figure 5(a). Then after perform Feature Extraction, Recognition using Bayesian decision theory as shown in Figure5(b).

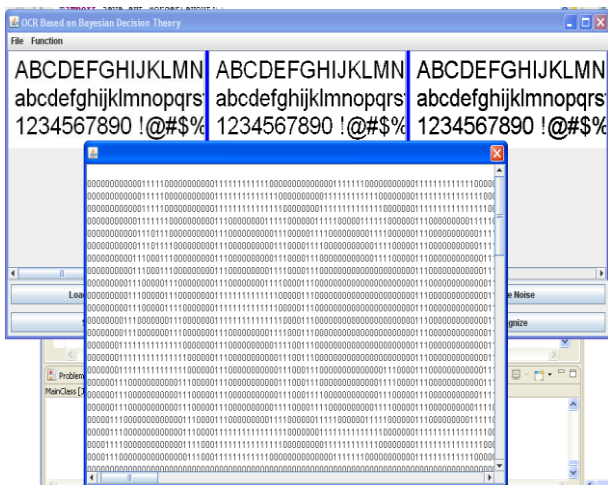


Figure 5(a) Binarization, Noise Remove and Segmentation

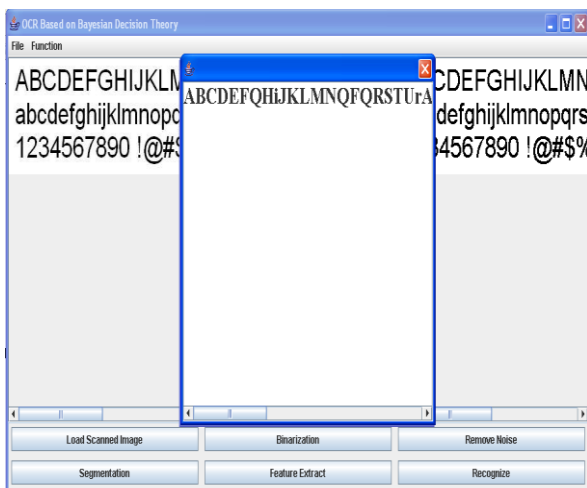


Figure 5(b) Recognition using Bayesian decision theory

5. Results and Discussion

This database contains 86,272 word instances from an 11,050 word dictionary written down in 13,040 text lines. We used the sets of the benchmark task with the closed vocabulary IAM-OnDB-t13. There the data is divided into four sets: one set for training; one set for validating the Meta parameters of the training; a second validation set which can be used, for example, for optimizing a language model; and an independent test set. No writer appears in more than one set. Thus, a writer independent recognition task is considered. The size of the vocabulary is about 11K. In our experiments, we did not include a language model. Thus the second validation set has not been used.

Table1. Shows the results of the four individual recognition systems [17]. The word recognition rate is simply measured by dividing the number of correct

recognized words by the number of words in the transcription.

We presented a new Bayesian decision theory for the recognition of handwritten notes written on a whiteboard. We combined two off-line and two online recognition systems. To combine the output sequences of the recognizers, we incrementally aligned the word sequences using a standard string matching algorithm. Evaluation of proposed Bayesian decision theory with existing recognition systems with respect to graph is shown in figure 6.

Table 1. Results of four individuals recognition systems

System	Method	Recognition rate	Accuracy
1st Offline	Hidden Markov Method	66.90%	61.40%
1st Online	ANN	73.40%	65.10%
2nd Online	HMM	73.80%	65.20%
2nd Offline	Bayesian Decision theory	75.20%	66.10%

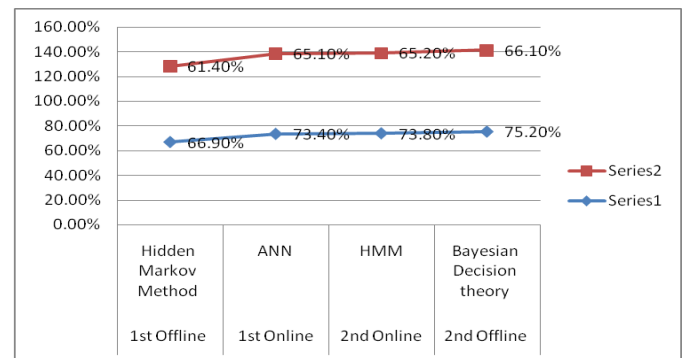


Figure 6 Evaluation of Bayesian decision theory with existing recognition systems

Then each output position the word with the most occurrences has been used as the final result. With the Bayesian decision theory could statistically significantly increase the accuracy.

6. Conclusion

We conclude that the proposed approach for offline character recognition, which fits the input character image for the appropriate feature and classifier according to the input image quality. In existing system missing characters can't be identified. Our approach using Bayesian Decision Theories which can classify missing data effectively which decrease error in compare to hidden Markova model. Significantly increases in accuracy levels will found in our method for character recognition

References

[1] El-Sheikh and R. M. Guindi, "Computer recognition of Arabic cursive scripts," Pattern Recognition., vol. 21, no. 4, pp. 293–302, 1988.

- [2] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber "A Novel Connectionist System for Unconstrained Handwriting Recognition" *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, may 2009
- [3] C. Y. Suenf, C. C. Tappert, and T. Wakahara, "The state of the art in on-line handwriting recognition," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 12, pp. 787–808, Aug. 1990.
- [4] L. Lam, S. W. Lee, and C. Y. Suen, "Thinning methodologies—A comprehensive survey," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 14, pp. 869–885, Sept. 1992
- [5] R. Plamondon and S. Sridhar. "On-line and Offline Handwriting Recognition: A Comprehensive Survey." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.
- [6] Mujtaba Husnain, Shahid Naweed, "English Letter Classification Using Bayesian Decision Theory and Feature Extraction Using Principal Component Analysis" *ISSN 1450-216X Vol.34 No.2*, pp.196-203, 2009.
- [7] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 690–706, July 2001.
- [8] J. Serra, "Morphological filtering: An overview," *Signal Process.*, vol.38, no. 1, pp. 3–11, 1994.
- [9] C. Downtown and C. G. Leedham, "Preprocessing and presorting of envelope images for automatic sorting using OCR," *Pattern Recognition.*, vol. 23, no. 3–4, pp. 347–362, 1998.
- [10] W. Guerfaii and R. Plamondon, "Normalizing and restoring on-line handwriting," *Pattern Recognition.*, vol. 26, no. 3, pp. 418–431, 1993.
- [11] Ø. D. Trier, A. K. Jain, and T. Text, "Feature extraction method for character recognition—A survey," *Pattern Recognition.*, vol. 29, no. 4, pp. 641–662, 1996.
- [12] R. Munguia, K. Tosca no, G Sanchez, M. Nakano "New Optimized Approach for Written Character Recognition Using Symlest Wavelet" *52nd IEEE International Midwest Symposium on Circuits and Systems 2009*.
- [13] L. O’Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Pattern Anal. Machine Intell.* vol. 15, pp. 162–173, 1993.
- [14] M. Y. Chen, A. Kundu, and J. Zhou, "Off-line handwritten word recognition using a hidden Markov model type stochastic network," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 481–496, May 1994.
- [15] I. S. Oh, J. S. Lee, and C. Y. Suen, "Analysis of class separation and combination of class-dependent features for handwriting recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 1089–1094, Oct. 2002
- [16] M. Sonka, V. Hlavac, and R. Boyle, "Image Processing, Analysis and Machine Vision", 2nd ed. Pacific Grove, CA: Brooks/Cole, 1999.
- [17] Marcus Liwicki, Horst Bunke "Combining On-Line and Off-Line Systems for Handwriting Recognition" *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*.

A Relationship Oriented Framework for Learning in Structured Domains

Madhusudan Paul, Thamizh Selvam. D, P. Syam Kumar and Dr. R. Subramanian

Department of Computer Science, School of Engineering and Technology,
Pondicherry University, Puducherry, India
{msp.cse@gmail.com, dthamizhselvam@gmail.com, shyam.553@gmail.com and rsmnian.csc@pondiuni.edu.in}

Abstract: Most of the classical machine learning (ML) models was developed to deal with non-structured domains of learning where data in input domain is represented as fixed size vector of properties. This kind of representation cannot always capture the true nature of the data which are naturally represented in some structured form like sequences and trees. Learning in structured domains (SDs) is a new field of study which allows for a generalization of ML approaches to the treatment of complex data, offering both new impulses for theory and applications. Since relationships are the key in SD learning, edges (relationships) are more important than vertices (indivisible component of objects) in SD learning. In the existing framework for graph processing of artificial neural network models, vertices are given more importance than edges. Again geometrical information of structured data is not considered in the framework. In this paper, a new framework for graph processing is proposed in which edges are considered as key and also geometrical information is taken into account.

Keywords: Artificial Neural Networks, Cascade Correlation Learning, Feedforward Neural Networks, Learning in Structured Domains, Recursive Neural Networks.

1. Introduction

In traditional machine learning, an input object in the input domain is represented by a fixed-size vector of properties (or features). Though this kind of representation is quite easy to realize and process, sometimes it cannot completely capture the “true nature” of the data which naturally presents itself in a structured form, since some important contextual information may be associated with the structure of the data itself. While learning in structured domains is quite difficult to realize and process, it is the generalized machine learning approach to deal with complex data structures. A non-structured domain can be thought as a special or restricted case of structured domains. A domain of real valued vectors can be treated as a special case of a domain of sequences of real valued vectors (i.e., it is a domain of sequences, all of size one), which in turn can be considered as a special case of a domain of trees (i.e., a sequence can be treated as a tree where all the internal nodes have a single child), which in turn can be considered as a special case of a domain of directed acyclic graphs (DAGs), and so on. Therefore, focusing on structured domains, does not exclude the possibility to exploit the traditional vector-based learning models. In fact, traditional vector-based approaches are just specific instances of a more general structure-based framework.

In traditional machine learning approaches, graphs or trees are mapped into simpler representations, like vectors. However the performances of these approaches differ largely with the application at hand. In fact, the preprocessing phase

is quite problem dependent and the implementation of this approach usually requires a time-consuming trial and error procedure. Moreover, the inherent topological information contained in structural representations might be partially lost.

Recently, new connectionist models, capable of directly elaborating trees and graphs without a preprocessing phase were proposed [1]. These have been extended using support vector machines [2]–[5], recursive neural networks [6] – [13] and SOMs [14] to structured data.

In the family of recursive neural networks (RNNs), constructive approach, recursive cascade correlation (RCC), has been introduced in [6]. RNN models realize an adaptive processing (encoding) of recursive (hierarchical) data structures. A recursive traversal algorithm is used to process (encode) all the graph vertices, producing state variable values for each visited vertex. In fact, all of these approaches can handle only sub-classes of graphs, not general graphs [i.e., rooted trees, directed acyclic graphs (DAG), and directed positional acyclic graphs (DPAG)]. Very recently as a first attempt, Neural Networks for Graphs (NN4G) [16] introduced the concept to the treatment of general class of graphs. It is an incremental approach where state values of vertices are updated gradually using cascade correlation learning algorithm. The principle idea behind the framework of the models is to obtain a flat description of the information associated to each vertex of graphs. In our proposed framework the idea is to obtain a flat description of the information associated to each edge instead of each vertex of graphs.

The rest of the paper is organized as follows. Section 2 introduces the preliminaries and notations on the domains. In Section 3, existing framework for graph processing is discussed. Our new proposed framework is explained in Section 4. The proposed framework is again enhanced in Section 5 considering geometrical information. Finally, future directions and conclusion is given in Section 6.

2. Preliminaries and notations

A labeled graph (or graph) g is a quadruple (V, E, L_v, L_e) , where V is the nonempty finite set of vertices/nodes, and E is the finite set of edges: $E \subseteq \{(u, v) | u, v \in V\}$. The last two items L_v and L_e associate a vector of real numbers to each vertex and edge, respectively. In fact, L_v and L_e are mappings as $L_v : V \rightarrow \mathbb{R}^{d_v}$ and $L_e : E \rightarrow \mathbb{R}^{d_e}$, where \mathbb{R}^{d_v} and \mathbb{R}^{d_e} are the sets of vectors of real numbers with dimension d_v and d_e , respectively. Vertex labels and edge labels are denoted by l_v , and $l_{(u,v)}$ (or l_e) respectively. The symbol l with no further specification represents the vector obtained by stacking together all the labels of the graph. The

symbol $|\cdot|$ denotes the cardinality or the absolute value, according to whether it is applied to a set or to a number.

If g is directed graph, each edge (u, v) of g is an ordered pair of vertices, where v is the children or successor of the parent or predecessor u . For undirected graph, the ordering between u and v in (u, v) is not defined, i.e., $(u, v) = (v, u)$. Vertex v and edge e are said to be incident with (on or to) each other, if vertex v is an end vertex of edge e . The number of edges incident on a vertex v with self-loops counted twice is called the degree of vertex v and denoted by $d(v)$. A cycle is a finite alternating sequence of vertices and edges beginning and ending with same vertex such that each edge is incident with the vertices preceding and following it and no edge and vertex (except initial and final vertex) are repeated. A graph with no cycle is called an acyclic graph.

Given a set of labeled graphs G and a graph $g \in G$, we denote the set of vertices of g as $V(g)$ and the set of its edges as $E(g)$. Given a vertex v , the vertices adjacent to v (or neighbors of v) are those connected to it by an edge and are represented by $N(v)$, i.e. $N(v) = \{u \in V(g) | (u, v) \in E(g)\}$. Hence, $d(v) = |N(v)|$.

Similarly, given an edge e , the edges adjacent to e (or neighbors of e) are those edges having a common end vertex with the given edge and are denoted by $N(e)$, i.e., $N(e) = \{(u, v) \in E(g) | u \in adj(e) \text{ or } v \in adj(e)\}$, where $adj(e)$ is the set of two end vertices of e .

If the graph is directed, the neighbors of v (or e), either belong to the set of its children or successors $S(v)$ (or $S(e)$) or to the set of its parents or predecessors $P(v)$ (or $P(e)$).

A graph is said to be positional if a function π is defined for each vertex v and assigns a different position $\pi(u)$ to each neighbor $u \in N(v)$, otherwise the graph is non-positional. Thus, combining the properties described so far it is possible to specify various graph categories: Directed Acyclic Graphs (DAGs), Directed Positional Acyclic Graphs (DPAGs) and so on.

Here we assume a class of input structured patterns as labeled graphs. Let a target function τ be defined as $\tau: g \times v \rightarrow \mathbb{R}^m$ (or $\tau: g \times e \rightarrow \mathbb{R}^m$), i.e., τ maps a graph g and one of its vertex v (or edge e) into a vector of real numbers. Our objective is to approximate the target function τ . More precisely, in classification problems codomain of τ is \mathbb{N}^m (i.e., vectors of natural numbers), whereas in regression problems the codomain is \mathbb{R}^m . In graph classification, τ does not depend on v (or e), i.e. only one target is given for each graph. In vertex (or edge) classification problems, each vertex (or edge) in a given set has a target to be approximated.

In this paper we face the problem of devising neural network architectures and learning algorithms for the classification of structured patterns, i.e., labeled graphs. Fig. 1 reports the standard way to approach this problem using a standard neural network. Each graph is encoded as a fixed-size vector which is then given as input to a feedforward neural network for classification. This approach is motivated by the fact that neural networks only have a fixed number of input units while graphs are variable in size. The encoding process is usually defined in advance and does not depend on the classification task. It is a very expensive trial and error approach.

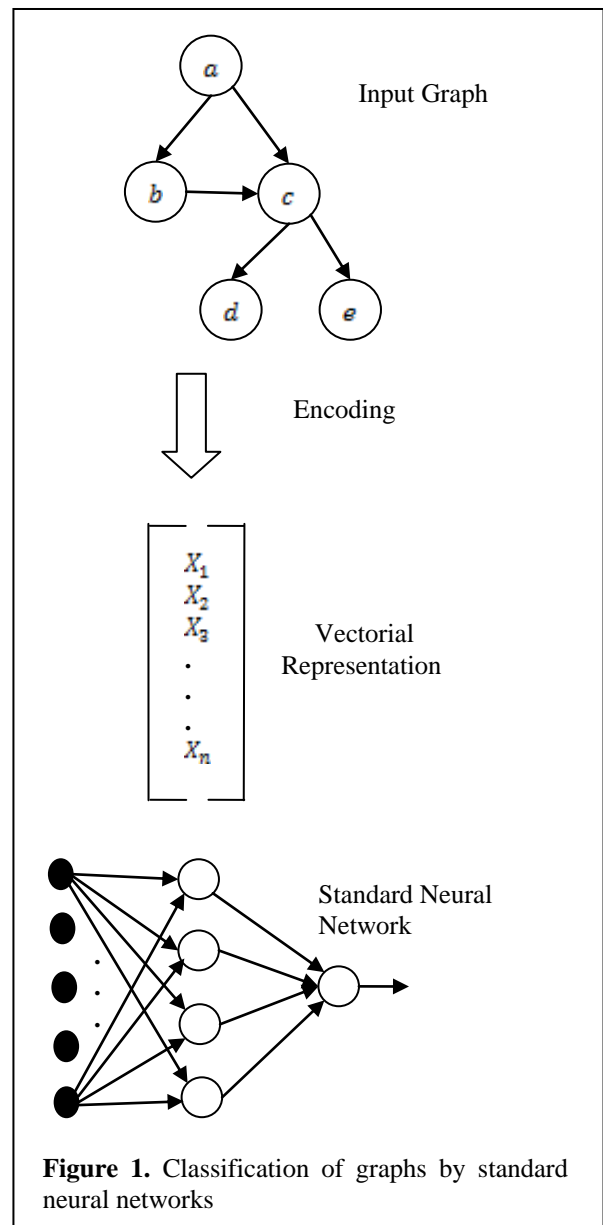


Figure 1. Classification of graphs by standard neural networks

Standard and recurrent neurons are not suitable to deal with labeled structures. In fact, neural networks using this kind of neurons can deal with approximation and classification problems in structured domains only by using a complex and very unnatural encoding scheme which maps structures onto fixed-size unstructured patterns or sequences. To solve this inadequacy of standard and recurrent neural networks the *generalized recursive neuron* was proposed in [6].

The generalized recursive neuron is an extension of the recurrent neuron where instead of just considering the output of the unit in the previous time step, we consider the outputs of the unit for all the vertices which are pointed by the current input vertex.

3. General Framework for Graph Processing

To define a general framework for graph processing, we need to implement a function $\varphi: g \times v \rightarrow \mathbb{R}^m$ to compute an output $\varphi(g, v)$ for each pair (g, v) . The principle idea is to derive a flat description of the information associated to each vertex v . An object of the domain of interest can be

represented by a vertex and its description is represented by a vector of real numbers called *state* denoted by $x_v \in \mathbb{R}^s$, where the *state dimension* s is a predefined parameter. In order to obtain a distributed and parallel processing framework, the *states* are computed locally at each vertex. A reasonable choice is to design x_v as the output of a parametric state transition function f_t , that depends on the vertex label l_v and on the relationships between v and its neighbors

$$x_v = f_t(l_v, x_{N(v)}, l_{N(v)}, l_{(v,N(v))}), \quad v \in V \quad (1)$$

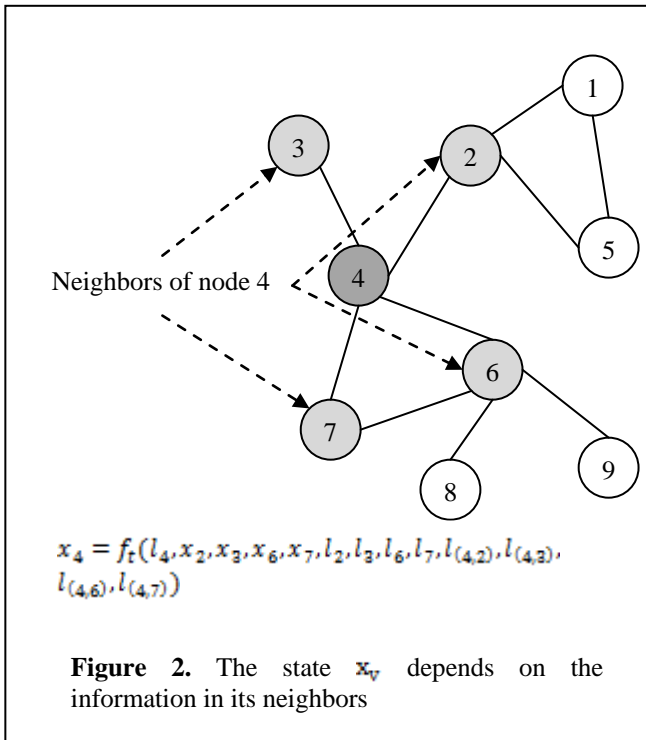
where $N(v)$ is the set of neighbors of vertex v , $x_{N(v)}$ and $l_{N(v)}$ are the sets containing the states and the labels of the vertices in $N(v)$ respectively, and $l_{(v,N(v))}$ is the set of the edge labels between v and its neighbors (Fig. 2).

Once each node has a vectorial representation, it can also be assigned an output y_v , evaluated by another parametric function g_t , called *output function*

$$y_v = g_t(x_v, l_v), \quad v \in V. \quad (2)$$

Eqns. (1) and (2) define a method to produce an output $y_v = \varphi(g, l_v)$ for each vertex of the graph g .

Moreover, the symbolic and subsymbolic information associated with the vertices is indeed automatically encoded into a vector by the state transition function. We can show the computation graphically substituting all of the vertices with “units” that compute the function f_t . The “units” are connected according to graph topology. The resultant network is called the *encoding network* and will be the same topology as the input graph. Since the same parametric functions are applied to all the vertices, the units of the same type share the same set of parameters.



Let F_t and G_t be the vectorial functions obtained by stacking all the instances of f_t and g_t , respectively. Then Eqns. (1) and (2) can be rewritten as

$$x = F_t(x, l), \quad (3a)$$

and

$$y = G_t(x, l), \quad (3b)$$

where l represents the vector containing all the labels and x collects all the states. Eq. (3a) defines the global state x , while Eq. (3b) computes the output. It is relevant to mention that Eq. (3a) is recursive with respect to the state x , thus x is well defined only if Eq. (3a) has a unique solution. In conclusions, the viability of the method depends on the particular implementation of the transition function f_t .

In our supervised framework, for a subset $S \subseteq V$ of vertices, called supervised vertices, a target value t_v is defined for each $v \in S$. Thus an error signal (usual sum of squared error) can be specified as,

$$e_t = \sum_{v \in S} (t_v - \varphi(g, v))^2. \quad (4)$$

This signal drives an error backpropagation procedure that adapts the parameters of f_t and g_t so that the function realized by the network can approximate the targets, i.e. $\varphi(g, v) \approx t_v \in S$.

In practice, Eq. (1) is well suited to process positional graphs, since each neighbor position can be associated to a specific input argument of function f_t . In non-positional graphs, this scheme introduces an unnecessary constraint, since neighbors should be artificially ordered. A reasonable solution consists in calculating the state x_v as a sum of “contributions”, one for each of its neighbors. Thus, state transition function can be rewritten as

$$x_v = \sum_{i=1}^{|N(v)|} h_t(l_v, x_{N_i(v)}, l_{N_i(v)}, l_{(v,N_i(v))}), \quad v \in V \quad (5)$$

where $N_i(v)$ is the i -th neighbor and $|N(v)|$ is the number of neighbors of v . Several possible implementations of the functions f_t (or h_t) and g_t can be selected, e.g., Recursive Neural Networks(RNNs), Graph Neural Networks(GNNs), and recently developed NN4G [16]. RNNs, GNNs and NN4Gs differ in the implementation of the state transition function f_t and in the class of graphs that can be processed.

4. Relationship oriented framework for Graph Processing

In the previous framework for graph processing, vertices are given more importance, i.e., state value of a vertex is computed based on the information associated with it and its neighbors. In our new proposed framework, state value is computed for each edge (instead of each vertex) on the information associated with it and its neighbors.

To define edge based framework for graph processing we must implement the function $\psi: g \times e \rightarrow \mathbb{R}^s$ instead of $\varphi: g \times v \rightarrow \mathbb{R}^s$ to compute an output $\psi(g, e)$ for each pair (g, e) . The principle idea behind this is to obtain a flat description of the information associated to each edge e instead of each vertex v . The description of an edge can be represented by a *state* value denoted by $x_e \in \mathbb{R}^s$, where the

state dimension s is a predefined parameter. To obtain a distributed and parallel processing framework, the states are computed locally at each edge. The state value x_e can be designed as the output of a parametric state transition function f_r (instead of f_t), which depends on the edge label l_e and on the labels of vertices adjacent to the edge and its neighbors

$$x_e = f_r(l_e, x_{N(e)}, l_{N(e)}, l_{adj(e)}), \quad e \in E \quad (6)$$

where $N(e)$ is the set of neighbors of edge e , $x_{N(e)}$ and $l_{N(e)}$ are the sets containing the states and the labels of the edges in $N(e)$ respectively, and $l_{adj(e)}$ is the set containing the labels of adjacent vertices of the edge e (Fig. 3).

Again each edge has a vectorial representation, it can also be assigned an output y_e , evaluated by another parametric function g_r (instead of g_t), called output function

$$y_e = g_r(x_e, l_e), \quad e \in E. \quad (7)$$

Eqns. (6) and (7) define a method to produce an output $y_e = \psi(g, l_e)$ for each edge of the graph g .

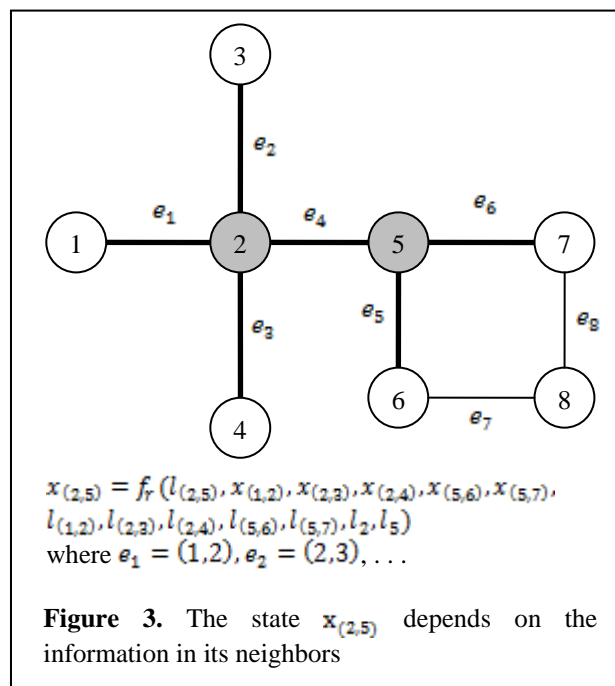
The symbolic and sub-symbolic information associated with the edges are automatically encoded into a vector by the state transition function f_r . The encoding network will be the same topology as the input graph. The units of the same type share the same set of parameters, since the same parametric functions are applied to all the edges.

Similar to vertex based framework, we can define the functions F_r and G_r by stacking all the instances of f_r and g_r , respectively and Eqns. (6) and (7) can be rewritten as

$$x = F_r(x, l) \quad (8a)$$

$$y = G_r(x, l) \quad (8b)$$

where l represents the vector containing all the labels and x collects all the states. Eq. (8a) defines the global state x , while Eq. (8b) computes the output. Note that Eq. (8a) is recursive with respect to the state x , thus x is well defined only if Eq. (8a) has a unique solution. The method differs from one implementation of the transition function (f_r) to another.



In the supervised framework, for a subset $S \subseteq E$ of edges, called supervised edges, a target value t_e is defined for each $e \in S$. Thus an error signal can be specified as,

$$\varepsilon_r = \sum_{e \in S} (t_e - \psi(g, e))^2. \quad (9)$$

This signal drives an error backpropagation procedure that adapts the parameters of f_r and g_r so that the function realized by the network can approximate the targets, i.e., $(g, e) \approx t_e \in S$.

A reasonable solution (like previous framework) consists in calculating the state x_e as a sum of “contributions”, one for each of its neighbors. Thus, state transition function can be rewritten as

$$x_e = \sum_{i=1}^{|N(e)|} h_r(l_e, x_{N_i(e)}, l_{N_i(e)}, l_{adj(e)}), \quad e \in E \quad (10)$$

where $N_i(e)$ is the i -th neighbor and $|N(e)|$ is the number of neighbors of e . New neural network models like RNNs, GNNs can also be developed to implement the functions f_r (or h_r) and g_r .

5. Learning in Geometrical Structured Domains

There are certain application domains where the “true nature” of the data not only depends on the hierarchical relationship of the data but also on the geometrical structure/topology of the data which naturally presents itself in a geometrical structured form and some contextual information is associated with the geometrical structure of the data itself. Quantitative structure-property/activity relationships (QSPR/QSAR) are fundamental aspects in chem.-informatics, where the aim is to correlate chemical structure of molecules with their properties (or biological activity) in order to achieve prediction. Since each molecule is a 3-D geometrical structure, the 3-D geometrical structure itself should have some impact on QSPR/QSAR. If we can develop a model that can learn geometrical topology, then the accuracy of QSPR/QSAR analysis of chemical

compounds may be improved more. We can find other applications also where learning in geometrical structured domains (GSD) can be incorporated to get better performance.

In Eq. (1), the label l_v of vertex v represents the information associated with the corresponding indivisible component object v of structured data. In fact, l_v is a fixed size vector of real numbers where each element of l_v represents a property or feature of the component object. Each component object itself is an unstructured (indivisible) data whereas these component objects are interconnected among them to form a structured data as a whole.

In both the frameworks of graph processing, geometrical information of structured data is not incorporated to compute the state values of vertices, which is essential to capture the true nature of geometrical structured data; in fact, only hierarchical nature of structured data is captured. We can incorporate relative coordinate information (if available) of each component object of structured data to calculate the state value of the corresponding vertex. Hence, Eq. (1) could be changed as

$$x_v = f_t(l_v, c_u, c_v, x_{N(v)}, l_{N(v)}, l_{adj(v)}), v \in V \quad (11)$$

and Eq. (6) could be rewritten as

$$x_e = f_r(l_e, c_u, c_v, x_{N(e)}, l_{N(e)}, l_{adj(e)}), e \in E \quad (12)$$

where $e = (u, v)$ and c_u, c_v are the vectors of coordinate of vertices u and v respectively. Based on dimension or size of vector c_u and/or c_v we can generalize the model into n - dimensional geometrical structured data. If input domain of data is 2-D geometrical structure, the size of c_u and/or c_v will be 2. Similarly, the size of c_u and/or c_v will be 3 when the data of input domain is 3-D geometrical structure.

6. Future Directions and Conclusion

Though the new framework proposed in this paper, can capture the true nature of structured data, the success of the framework still lie on the proper implementation of the framework. Hence further research could be done on the development of new artificial network models.

Again the development of model for learning in geometrical structured domains (GSD) needs to investigate on the integration of symbolic and sub-symbolic approaches. The integration of symbolic and sub-symbolic approaches is a fundamental research topic for the development of intelligent and efficient systems capable of dealing with tasks whose nature is neither purely symbolic nor sub-symbolic. It is common opinion in the scientific community that a extensive variety of real-world problems require hybrid solutions, i.e., solutions combining techniques based on neural networks, fuzzy logic, genetic algorithms, probabilistic networks, expert systems, and other symbolic techniques. A very popular view of hybrid systems is one in which numerical data are processed by a sub-symbolic module, while structured data are processed by the symbolic counterpart of the system. Unfortunately, because of the different nature of numerical and structured representations, a tight integration of the different components seems to be very difficult.

Learning in structured domains can be used in various fields of applications such as natural language processing, image processing, speech processing, computer vision, chem.-informatics, bioinformatics, etc. Hence, a simple solution for general class of graphs is anticipated. The NN4G model is a relatively simple solution for dealing with fairly general classes of graphs by sub-symbolic approaches; similar solution for geometrical structures is also expected. We hope that the introduction of simple and general approaches (e.g., NN4G) in structured domains will be attracted to ML researchers for widespread applications.

References

- [1] B. Hammer and J. Jain, "Neural methods for non-standard data," in Proceedings of the 12th European Symposium on Artificial Neural Networks, M.Verleysen, Ed., 2004, pp. 281–292.
- [2] R. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in Proc. 19th International Conference on Machine Learning (ICML2002), C. Sammut and A. e. Hoffmann, Eds. Morgan Kaufmann Publishers Inc, 2002, pp. 315–322.
- [3] T. Gärtner, "Kernel-based learning in multi-relational data mining," ACM SIGKDD Explorations, vol. 5, no. 1, pp. 49–58, 2003.
- [4] P. Mahé, N. Ueda, T. Akutsu, P. J.-L., and J.-P. Vert, "Extensions of marginalized graph kernels," in Proc. 21th International Conference on Machine Learning (ICML2004). ACM Press, 2004, p. 70.
- [5] J. Suzuki, H. Isozaki, and E. Maeda, "Convolution kernels with feature selection for natural language processing tasks," in ACL, 2004, pp. 119–126.
- [6] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," IEEE Transactions on Neural Networks, vol. 8, pp. 429–459, 1997.
- [7] P. Frasconi, M. Gori, and A. Sperduti, "A general framework for adaptive processing of data structures," IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 768–786, 1998.
- [8] M. Bianchini, M. Gori, and F. Scarselli, "Processing directed acyclic graphs with recursive neural networks," IEEE Trans. Neural Netw., vol. 12, no. 6, pp. 1464–1470, Nov. 2001
- [9] A. Micheli, D. Sona, and A. Sperduti, "Contextual processing of structured data by recursive cascade correlation," IEEE Trans. Neural Netw., vol. 15, no. 6, pp. 1396–1410, Nov. 2004.
- [10] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in Proc. International Joint Conference on Neural Networks (IJCNN2005), 2005, pp. 729–734.
- [11] F. Scarselli, S. Yong, M. Gori, M. Hagenbuchner, A. Tsoi, and M. Maggini, "Graph neural networks for ranking web pages," in Proc. of the 2005 IEEE/WIC/ACM Conference on Web Intelligence (WI2005), 2005, pp. 666–672.
- [12] B. Hammer, A. Micheli, and A. Sperduti, "Universal approximation capability of cascade correlation for structures," Neural Comput., vol. 17, no. 5, pp. 1109–1159, 2005.
- [13] M. Bianchini, M. Gori, L. Sarti, and F. Scarselli, "Recursive processing of cyclic graphs," IEEE Trans. Neural Netw., vol. 17, no. 1, pp. 10–18, Jan. 2006.

- [14] M. Hagenbuchner, A. Sperduti, and A. C. Tsoi, "A self-organizing map for adaptive processing of structured data," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 491–505, May 2003.
- [15] M. Bianchini, M. Maggini, L. Sarti, and F. Scarselli, "Recursive neural networks for processing graphs with labelled edges: Theory and applications," *Neural Networks - Special Issue on Neural Networks and Kernel Methods for Structured Domains*, vol. 18, pp. 1040–1050, October 2005.
- [16] Alessio Micheli, "Neural Network for Graphs: A Contextual Constructive Approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, March 2009.
- [17] S. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," *Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-90-100*, Aug. 1990.
- [18] A. Kuchler and C. Goller, "Inductive learning in symbolic domains using structure-driven recurrent neural networks," in *Advances in Artificial Intelligence*, G. Goro and S. H. Oldobler, Eds. Berlin: Springer-Verlag, 1996, pp. 183–197.
- [19] V. Di Massa, G. Monfardini, L. Sarti, F. Scarselli, M. Maggini, and M. Gori, "A comparison between recursive neural networks and graph neural networks," in *Proc. World Congr. Comput. Intell./Int. Joint Conf. Neural Netw.*, 2006, pp. 778–785.
- [20] Ethem Alpaydin, "Introduction to Machine Learning," Prentice-Hall of India, 2005.
- [21] Martin T. Hagan, Howard B. Demuth, Mark Beale, "Neural Network Design," Cengage Learning India, 1996.
- [22] Simon Haykin, "Neural Networks, A Comprehensive Foundation," 2nd ed. Pearson Education, 2006.
- [23] Mohamad H. Hassoun, "Fundamentals of Artificial Neural Networks," Prentice-Hall of India, 2008.
- [24] Duane Hanselman, Bruce Littlefield, "MASTERING MATLAB 7," Pearson Education, 2005.
- [25] Jue Wang and Qing Tao, "Machine Learning: The State of the Art," *IEEE Intelligent Systems*, 2008.

Author Biographies

MADHUSUDAN PAUL He is a final year student of M.Tech. (CSE) in the Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India. He received his M.Sc. (CS) in 2008 from Visva-Bharati University, Santiniketan, West Bengal, India and B.Sc. (CS) in 2006 from University of Calcutta, Kolkata, West Bengal, India. His research interests include Machine Learning, Soft Computing, and Automata Theory.

THAMIZH SELVAM. D He received his B.Sc., Computer Science and M.Sc., Computer Sciences from Pondicherry University, Puducherry, India in 2000 and 2003 respectively. He received his M.Phil., in Computer Science from Periyar University, Tamilnadu, India in 2008. Currently, he is pursuing his Ph.D., in Computer Science and Engineering from Department of Computer Science, School of Engineering, Pondicherry University, Puducherry. His field of research includes Distributed Algorithms, Peer-to-Peer Networks, and Overlay Network Structures.

P. SYAM KUMAR He received his B.Tech., (CSE) from JNTU, Hyderabad, India and M.Tech., (CST) from Andhra University, Visakhapatnam, India in 2003 and 2006 respectively. His research interests include Distributed Computing and Cloud Computing.

Dr. R. SUBRAMANIAN He is currently the Professor and Head of Department of Department of Computer Science, School of Engineering, Pondicherry University, Puducherry, India. He received his B.Sc., Mathematics from Madurai Kamaraj University, Madurai, India in 1982. He received his M.Sc., Mathematics and Ph.D., in Computer Science and Engineering from IIT Delhi in 1984 and 1989. He has in his credit around 20 research publications in peer-scholarly research publications in both National and International Journals and Conferences. His research interests include Parallel and Distributed Algorithms, Evolutionary Algorithms and Robotics

Is Service Discovery necessary and sufficient – A Survey

K.V. Augustine, E.Rajkumar,

Department of Computer Science
Pondicherry University
Puducherry
augustine.k.v@gmail.com

Abstract— Service computing is a cross discipline technology emerged to fill the gap between the information technology and business process. A Service is an operation or set of operation provided by an entity to another entity through contracted interface. The success of its utility lies on its capability to abide to the requirement of its consumer. The service is said to be optimum if the preference of the puller is maximum or completely satisfied. In this paper we evaluate some of the approaches that have emerged for and technically termed as service discovery. Here we also provide the challenges that have to be addressed when defining the discovery process and how far they have been satisfied. We also discuss some of the Quality of Service (QoS) factors that are to be considered.

Keywords – Service Computing, Service Discovery, QoS.

1 Introduction

Service oriented computing (SOC) has gained momentum with the paradigm shift from proprietary standard to global standard. The overview of the service computing is given in fig.1. The origin of SOC is from vast areas of computing such as component based, enterprises integration business modeling etc. But today it has gone far ahead of its origin into emergence of areas like Web2.0 applications, business process management, software as service, data as a service, and cloud computing. Still the need of technology to merge the business process and organization structure to find the organization cons and address them with a solution is enormous.

The research challenges that are faced today in the area are Dynamically (re-)configurable run-time architecture, Dynamic connectivity capabilities, Topic and content-based routing capabilities, End-to-end security solutions, Infrastructure support for application integration, Infrastructure support for data integration, Infrastructure support for process integration, Enhanced service discovery. Moreover, quality properties need to be addressed. The competitive push of the providers and competence pull of consumers has made this environment an ever green research area.

In this paper we analyze the approaches in enhanced service discovery. As the number of services has increased the effort needed by the consumer to run through each service to find the service of his choice is taxing. To overcome this challenging an effective discovery mechanism is significant. The major challenge lies in digging out the optimal service precisely. The basic sources for discovery are repositories where the description and location of the service is depicted

and service descriptions like WSDL where the interface, operations input and output parameters are provided.

The remainder of this paper is organized as follows: Section 2 Challenges in service discovery 3 Service discovery taxonomy 4 survey on discovery 5 Quality of Service 6 Conclusion and future direction.

2 Challenges in Service Discovery

2.1 Preferred service tracing

Due to ubiquitous growth of services the major challenge lies in discovering the service that best suits to the user demand from among the available offer.

2.2 Uniform description

The specification of the provides and the goal demand of the consumer must have common standard so that the matching of the demand and offer can be made easier. The major challenges lie in developing a standard specification for both service and goal.

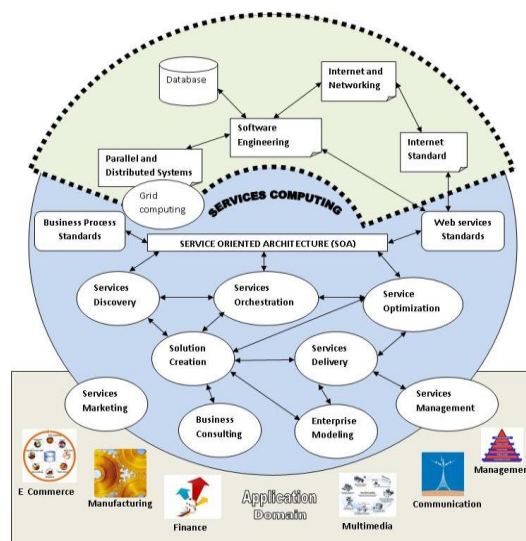


Fig1: Overview of Service Computing

2.3 Semantic annotation

The UDDI registry of a service support only keyword based search of the service. The underlying semantic of the demand or the offer may improve the efficiency of the discovering mechanism. A better annotation techniques that provides the underlying semantics is a major challenge in discovery

2.4 Clustering approach

Clustering plays a vital role in reducing the search space for discovery. Most of the clustering approach are based on the functional aspects of service that is similarity measure of input, output, precondition and effects. A better clustering method that can incorporate both functional and non functional approach is a major challenge in service discovery.

2.5 Mediator engine

The source of service description are heterogeneous in nature, a need for an mediator that can extract the description from the these heterogeneous sources and provide them for the discovery frame work is another challenging issues in service discovery.

2.6 Similar service discovery

Discovering set of services that are similar in nature to the service in hand is another challenging issue that has be addressed in recent year.

2.7 Storage model

Due to increase in number of services a service aggregator which acts a intermediate storage by clustering services based on functional and nonfunctional characteristics of service has became essential. A better storage model that can enable easy access and retrieval of service is a challenging issue.

2.8 Matching and Ranking

Matching is an important in aspect in service discovery. The challenge lies in providing a best matching method that matches the demand and the offer. Similar to matching ranking of service plays a vital is discovering the service that best suits the demand constrain. A ranking model based on quality is a challenging task in web service discovery.

3 Service discovery Taxonomy

In this section we discuss some the requirement and importance of service discovery together with some types and support for discovery. The overall taxonomy is depicted in fig 2.

3.1 Requirements of Service Discovery

We argue that the following requirements, over and above the generic requirements of services, are necessary to support service discovery in any context:

- Descriptions must be attached to different resources (services and workflows) published in different components (service registries, local file stores or databases)

- Publication of descriptions must be supported both for the author of the service and third parties
- Different classes of user will wish to examine different aspects of the available metadata, both from the service publisher
- There is a need for control over who make add and alter third party annotations
- We must support two types of discovery: the first using cross-domain knowledge; the second requiring access to common domain ontologies
- A single, unified interface for all these kinds of discovery should be made available to the user.

3.2 Importance of Service Discovery

To illustrate the importance of service discovery, the following impact shows the way for it. Alternatively one could assume to directly query the web service during the web service discovery process. However, this may lead to network and server overload and it makes a very strong assumption: in addition to data mediation, protocol and process mediation for the web service must be in place before the discovery process even starts. Without thinking that this is a realistic assumption, in consequence assumption is made that it is essential.

Taking the analogy with databases as illustration, web service discovery is about searching for databases that may contain instance data we are looking for, while service discovery is about finding the proper instance data by querying the discovered databases. The same analogy can be extended to consider services that imply some effects in the real world. Service discovery is about checking whether the ticket sellers offering such web services can really provide the concrete requested ticket. With this example, the importance of service discovery is known as clear crystal.

3.3 Support for retrieval of Service

3.3.1 Keyword-Based Retrieval:

Search based on keywords from the service request. This method is highly sensitive to the 'zero or a million' problem because keyword are a poor method to capture the semantics of a request. Keywords can be synonyms (i.e., syntactical different words can have the same meaning) or homonyms (i.e., equal words can have different meanings) leading to low precision and recall. Furthermore, the relationship between different keywords in a request cannot be handled.

3.3.2 Table-Based Retrieval

It consists of attribute value pairs that captures service properties (e.g., output = article name). Services and requests are both represented as tables with attribute-value pairs and then matched. Semantics are more precisely captured in this method than in keyword-based retrieval but still the problems with synonyms and homonyms exist.

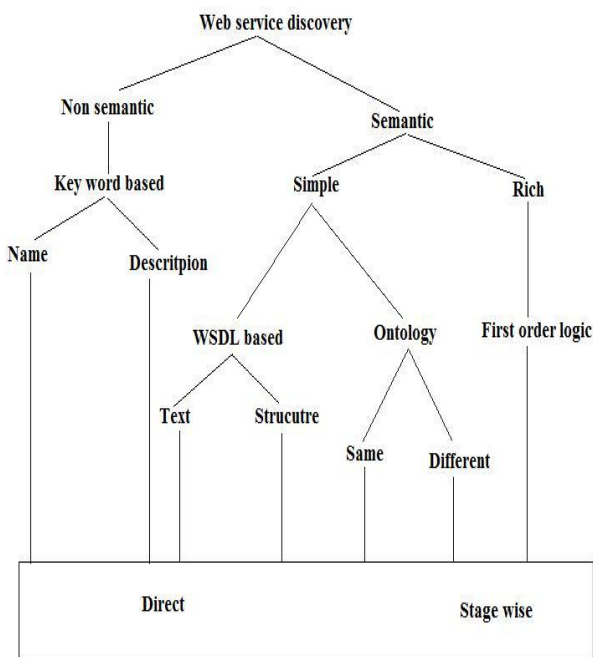


Fig 2: Service Discovery Taxonomy

3.3.3 Concept-Based Retrieval

Defines ontologies for classification of services thereby enabling retrieval on types rather than on keywords. This approach can give increased precision and recall difficult to develop and manage (i.e., difficult to define a consistent ontology of the world, how to combine ontologies with contradictory concepts).

3.3.4 Deductive Retrieval

In this approach, service semantics are expressed formally using logic. Retrieval then consists of deducing which service achieves the functionality described in the query. Theoretically, this method can achieve perfect recall and precision. The problems of this method are more practical. Formal modeling of service description and service request is very hard. Furthermore, the matching process in this method (proofing) can have high complexity and therefore operates slowly.

3.4 Types of Discovery

3.4.1 Goal Discovery

Users may describe their desires in a very individual and specific way that makes immediate mapping with service descriptions very complicated. Therefore, each service discovery attempt requires a process where user expectations are mapped on more generic goal descriptions. Notice that this can be hidden by the fact that a discovery engine allows the user only to select from predefined goals. However, then it is simply the user who has to provide this mapping i.e. who has to translate his specific requirements and expectations into more generic goal descriptions. This step can be called goal

discovery, i.e., the user or the discovery engine has to find a goal that describes (with different levels of accuracy) his requirements and desires.

3.4.2 Service Discovery

Service discovery is based on the usage of web services for discovering actual services. Web service technology provides automated interfaces to the information provided by software artifacts that is needed to find, select and eventually buy a real-world service or simply find the piece of information somebody is looking for.

3.4.3 Web Service Discovery

Web service discovery is based on matching abstracted goal descriptions with semantic annotations of web services. This discovery process can only happen on an ontological level i.e., it can only rely on conceptual and reusable.

3.5 Service Discovery Approaches

Service Discovery is done based on the following ways:

- Service Discovery based on Keyword Matching
- Service Discovery based on Simple Semantic Description of Service
- Service Discovery based on Rich Semantic Description of Services

3.5.1 Service Discovery based on Keyword Matching

It is a basic ingredient in a complete framework for semantic web service discovery. By making a keyword base search the huge amount of service can be filtered or ranked rather quickly. In a typical keyword based scenario a keyword based engine is used to discover services. A query, which is basically a set of keyword, is provided as input to the query engine. The query engine matches the keywords from to query against the keyword used to describe the services. A query with the same meaning can be formulated by using a synonyms dictionary, like Word-Net. The semantic of the query remains the same but because of the different keyword used, synonyms of previous ones, more services that possible fulfill user request are found. Moreover, by using dictionaries like WordNet as well as natural language processing techniques increases of the semantic relevance of search result.

3.5.2 Service Discovery based on Simple Semantic Description of Service

It uses the controlled vocabularies with explicit, formal semantics. Ontologies, which offer a formal explicit specification of a shared conceptualization of some problem domain, are excellent and prominent conceptual means for this purpose. They provide an explicit and shared terminology, explicate interdependencies between single concepts and thus are well suited for the description of web services and requestor goals. Moreover, Ontologies can be formalized in logics, which enable the use of inference service for exploiting

knowledge about the problem domain during matchmaking and discovery.

3.5.3 Service Discovery based on Rich Semantic Description of Services

It does the reasoning over the first order formulae in set based modeling, which leads to the extension of set based modeling. This extension of the set based modeling approach for web service to rich semantic descriptions, which capture the actual relationship between inputs and outputs/effects of web service execution as well and gave the formalization in first-order languages. Instead of considering the state of the world, here consider the service as input, output, precondition, assumptions, Post conditions and effects of services.

4 Survey on Discovery

4.1 Centralised and Decentralised registries

Discovery of Web services is of an immense interest and is a fundamental area of research in ubiquitous computing. Many researchers have focused on discovering Web services through a centralized UDDI registry [5-7]. Although centralized registries can provide effective methods for the discovery of Webservices, they suffer from problems associated with having centralized systems such as a single point of failure, and bottlenecks. In addition, other issues relating to the scalability of data replication, providing notifications to all subscribers when performing any system upgrades, and handling versioning of services from the same provider have driven researchers to find other alternatives

Other approaches focused on having multiple public/private registries grouped into registry federations [8,9] such as METEOR-S for enhancing the discovery process. METEOR-S [9] provides a discovery mechanism for publishing Web services over federated registry sources but, similar to the centralized registry environment, it does not provide any means for advanced search techniques which are essential for locating appropriate business applications. In addition, having a federated registry environment can potentially provide inconsistent policies to be employed which will significantly have an impact on the practicability of conducting inquiries across the federated environment and can at the same time significantly affect the productiveness of discovering Web services in a real-time manner across multiple registries.

4.2 Non Semantic approach

The growing number of web services available within an organization and on the Web raises is a challenging search problem: locating desired web services. In fact, to address this problem, several simple search engines have implemented [1, 2, 3, 4]. These engines provide only simple keyword search on web service descriptions. Keyword based search techniques do not consider the semantic description of services. Thus, they suffer from poor precision and recall.

Text-based method is the most straightforward way to conduct Web service discovery. The most widely used text-based is the keyword matching built in the UDDI public registry. The UDDI API allows developers to specify keywords of particular interests and it then returns a list of Web services whose service description contains those keywords. Beyond the literal keyword matching, research in XML schema matching[11] has applied various string comparison algorithms (e.g. *prefix*, *suffix*, *edit distance*) to match those interchangeable keywords but with slightly different spellings. Although keyword matching methods (i.e. broad, phrase, exact, and negative) may partially support the discovery of Webservices, they do not provide clients with efficient ways for articulating proper service queries (i.e. consider input/output values of service operations).

4.3 Semantic based discovery

A comprehensive description of the SWS is given in [12]. The fundamental idea underlying current SWS community is that in order to achieve machine-to-machine integration, a markup language (e.g. annotation) must be descriptive enough that a computer can automatically determine its meaning. Following this principle, many semantic annotation markup languages for Web services have come into existence and use such as OWL-S [13], (formerly known as DAML-S [14]), and WSDL-S [15] that have gained great momentum in recent years. The main goal of both OWL-S and WSDL-S is to establish a framework within which service descriptions are made and shared.

4.3.1 WSDL-based and Ontology-based.

The rationale is that WSDL is the defacto standard for representing a Web service's functional capability and technical specifications "on the wire" [16]. It is then natural to discern service discovery methods that centre upon WSDL with those do not. It should be noted that these two categories are not absolutely orthogonal with each other. For example, in the ontology-based method WSDL-S [17], annotation have been made to reference to a domain ontology through the standard WSDL extension mechanism. Hence, we define that WSDL based refers to those methods that take regular WSDL files 'as-is' without further augmenting. Ontology based methods[18], on the other hand, aim to provide a 'semantically enriched' version of WSDL files in order to automate complicated tasks such as service composition.

4.3.2 WSDL based approaches

In this section, we succinctly survey WSDL-based approaches In [19] VSM is used to build a Web service search engine. [20] has attempted to leverage LSA, a variant of VSM, to facilitate web services discovery. However, both [19] and [20] rely on existing UDDI public registries. In [22], a WSDL file is treated as a structural tree that can be compared based on the structures of the operations' input/output messages, which in turn, is based on the comparison of the data types delivered contained in these messages. Likewise, the interface similarity defined in [23] is computed by identifying the pair-wise correspondence of their operations that maximizes the sum

total of the matching scores of the individual pairs. The author in [21] calculated the similarity of complex WSDL concepts given similarity scores for their sub-elements. Using the maximum-weighted bipartite matching [24] algorithm from the graph theory, the author defined a number of coefficients to determine the ultimate structural similarity score between two parts in a matching pair. Most of these WSDL structural matching methods are inspired from the signature matching [25], a software component retrieval method from software engineering research. Although a standard WSDL does not provide semantic information, identifiers of messages and operations do contain information that can potentially be used to infer the semantics. When comparing two operations in [22], WordNet is used for deriving the synonyms for the semantic similarity calculation. The lexical similarity defined in [23] and [26] is also based on the concept distance computed from the WordNet sense hierarchy. Interestingly, research in [21] indicates that using WordNet may bring many false correlations due to its excessive generality.

4.4 Clustering in service discovery

More recently, clustering approaches are used for discovering Web services [27, 28, and 29]. Dong [29] puts forward a clustering approach to search Web services where the search consisted of two main stages. A service user first types keywords into a service search engine, looking for the corresponding services. Then, based on the initial Web services returned, the approach extracts semantic concepts from the natural language descriptions provided in the Web services. In particular, with the help of the co-occurrence of the terms appearing in the inputs and outputs, in the names of the operations and in the descriptions of Web services, the similarity search approach employs the agglomerative clustering algorithm for clustering these terms to the meaningful concepts. Through combination of the original keywords and the concepts extracted from the descriptions in the services, the similarity of two Web services can be compared at the concept level so that the proposed approach improves the precision and recall.

Arbrawowicz [28] proposes an architecture for Web services filtering and clustering. The service filtering is based on the profiles representing users and application information, which are further described through Web Ontology Language for Services (OWL-S). In order to improve the effectiveness of the filtering process, a clustering analysis is applied to the filtering process by comparing services with related clusters. The objectives of the proposed matchmaking process are to save execution time, and to improve the refinement of the stored data. Another similar approach [27] concentrates on Web service discovery with OWL-S and clustering technology, which consists of three main steps. The OWL-S is first combined with WSDL to represent service semantics before a clustering algorithm is used to group the collections of heterogeneous services together. Finally, a user query is matched against the clusters, in order to return the suitable services.

Another approach [30] focuses on service discovery based on a directory where Web services are clustered into the predefined hierarchical business categories. In this situation, the performance of reasonable service discovery relies on both service providers and service requesters having prior knowledge on the service organization schemes.

The approach of CPLSA [31] has some similarities to the older approaches [27, 28, and 29] in that keywords are used to first retrieve Web services, and extract semantic concepts from the natural language descriptions in the Web services. This work differs from other works in several ways. Firstly, we eliminate irrelevant service via exploiting a clustering algorithm to diminish the size of services returned; this approach shows some potential applications like over mobile uses. Secondly, based on the characteristics of Web services with a very limited amount of information, we regard the extraction of semantic concepts from service description as a problem of dealing with missing data. Therefore, this work utilizes Probabilistic Latent Semantic Analysis (PLSA) a machine learning method, to capture the semantic concept hidden behind the words in a query and the advertisements in services.

Another recent approach [32] is discovering homogeneous service communities through web service clustering. It gathers the features for a WSDL file is not as simple as collecting description documents when assuming no central UDDI registries. Another closely related area is the conventional document or web page clustering. They both involve the discovery of naturally-occurring groups of related documents. Web service files do not usually contain sufficiently large number of words for use as index terms or features. Moreover, the small numbers of words present in the web service files are erratic and unreliable. Hence, conventional, detailed linguistic analysis, and statistical techniques using local corpora cannot be applied directly for web service files clustering. The use of link analysis between WSDL files to discover related web services has also been studied. In our experiments, we employed Google API's search options for discovering web page referral or citation. However, it is discovered that most of the WSDL files do not have related web pages that provide hyperlinks to them. For the few that have hyperlinks referring to them, such WSDL files are typically educational examples for teaching how to program in a service-oriented paradigm.

In short, the individual existing techniques borrowed from related research areas such as information retrieval are inadequate for the purpose of discovering functionally-related web service clusters. While there is a small number of existing approaches dedicated to the discovery of web services as mentioned above, most of them remain hypothetical in nature, and have yet to be implemented and tested with real-world datasets.

4.5 Similar service discovery

In an approach [32] in discovering similar service operation where for a given service the service similar to services are discovered from the repositories by matching the input, output and operation of the service. The approach proposed schema matching techniques for the input, output datatypes schemas.

In another approach [33] Xion Dong et.al proposed a search engine woogle which searches for a similar service operation given an operation as input. In their approach the matching problem is consider similar to the text document matching problem, database schema matching problem and software component matching problem in their approach they match input, output as concepts by using A-prior algorithm for association rules and agglomeration algorithm for clustering similar concepts. In [34] a recent approach the similar approach as [33] was used with improvement in the clustering techniques using domain taxonomy. They also proposed a “service pool” to store similar service which may be single service or composite service using Graph based techniques.

5 Quality of Service attributes

Based on the behavior the quality of a service can be classified as given in [10]

Computational behavior: These QoS includes Execution Attributes (Latency, Accuracy, Throughput, Reliability, Extendibility, Capacity, and Exception Handling), Security (such as Encryption, Authentication, and Authorization), Secrecy, Availability etc.

Business behavior: These QoS mainly refers to Execution Cost, Reliability of the provided service, Punishment on condition that SLA could not be sufficed with.

Metadata restriction: These QoS includes Constraints that have to be followed regarding UDDI /WSDL /SOAP/WSLA parameters such as location, specific companies , schema .

5.1 Execution attributes

1. *Response Time*: time elapsed from the submission of a request to the time the response is received.
2. *Accessibility*: represents the degree that a service is able to serve a request.
3. *Compliance*: represents the extent to which a WSDL document follows WSDL specification
4. *Successability*: represents the number of request messages that have been responded.
5. *Availability*: represents the percentage of time that a service is operating.

5.2 Security

It is related to the ability of a given Web service to provide suitable security mechanisms by considering the following three parameters.

1. *Encryption*: the ability of a Web service to support the encryption of messages.
2. *Authentication*: the capacity of a Web service to offer suitable mechanisms dealing with the identification of the invoking party and allow operation invocation.

3. *Access control*: whether the Web service provides access control facilities to restrict the invocation of operation and the access to information to authorized parties.

5.3 Business attributes

Like QoS properties, they are relevant for differentiating Web services having the same functional characteristics

1. *Cost*: represents money that a consumer of a Web service must pay in order to use the Web service.
2. *Reputation*: measures the reputation of Web services based on user feedback
3. *Organization arrangement*: includes preferences and history (ongoing partnerships)
4. *Payment method*: represents the payment methods accepted by a Web service, i.e. transfer bank, Visa card etc.
5. *Monitoring*: required for a number of purposes, including performance tuning, status checking, debugging and troubleshooting.

6 Conclusion

Service Computing has gained momentum during the years and its proliferation has bridged the gap between the business and IT industries to make them work close to each other. The advent made has forced the challenges faced as given in section 1 to be addressed with more significance and enhanced its research scope. Due to the exponential growth of service and the relation between demand and supply made the aspects like service discovery more prominent. The work that has been undertaken in that has directions reveals the necessity of it. So necessity part has become vital. In sufficiency point of view does the promising approaches that are discussed in section 4 satisfies. The two measure that are commonly used to measure the efficiency of the approaches are precision and recall. Keyword based approaches is not unto the expectation more prominence is on semantic based approaches. The efficiency of the semantic based approaches has significantly increased as for the precision and recall are concerned, still the lack in optimality. More work has to done with the advent in technology to make the discovery process more and more optimal. Our survey above is to motivate the researcher to work on this area has this has become a promising area of research.

References

- [1]. Binding point. <http://www.bindingpoint.com/>.
- [2]. Grand central. <http://www.grandcentral.com/directory/>.
- [3]. Salcentral. <http://www.salcentral.com/>.
- [4]. Web service list. <http://www.webservicelist.com/>.
- [5]. U. Thaden, W. Siberski, and W. Nejd, “A semantic web Based Peer-to-Peer Service Registry Network,” TechnicalReport, Learning Lab Lower Saxony, 2003.

- [6]. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, "Semantic matching of web services capabilities," ISWC, pp. 333-347, 2002.
- [7]. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara "Importing the semantic web in UDDI," International Workshop on Web Services, E-Business, and the Semantic Web, pp. 225-236, 2002.
- [8]. K. Sivashanmugam, K. Verma, and A. Sheth, "Discovery of web services in a federated registry environment," ICWS, pp. 270-278, 2004.
- [9]. C. Zhou, L. Chia, B. Silverajan, and B. Lee, "UX- an architecture providing QoS-aware and federated support for UDDI," ICWS, pp. 171-176, 2003.
- [10]. Quality Model for Web Services v2.0 Committee Draft, September 2005
- [11]. H. H. Do and E. Rahm, "COMA - A system for flexible combination of schema matching approaches," presented at 28th VLDB Conference, Hong Kong, China, 2002.
- [12]. Processes and Applications," in *Semantic Web and Beyond: Computing for Human Experience*, R. Jain and A. Sheth, Eds.: Springer, 2006.
- [13]. D. Martin, "OWL-S: Semantic Markup for Web Services," in *Releases of DAML-S / OWL-S*, 2004.
- [14]. M. Paolucci, K. Sycara, T. Nishimura, and N. Srinivasan, "Using DAML-S for P2P Discovery," *Proceedings of First International Conference on Web Services, ICWS*, 2003.
- [15]. IBM and UGA, "Web Service Semantics," IBM and UGA 2005.
- [16]. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," 2001.
- [17]. "Web Service Semantics," [http://www.w3.org/ Submission /WSDL-S/](http://www.w3.org/Submission/WSDL-S/).
- [18]. J. Cardoso and A. Sheth, "Semantic Web Services, Processes and Applications," in *Semantic Web and Beyond: Computing for Human Experience*, R. Jain and A. Sheth, Eds.: Springer, 2006.
- [19]. C. Platzer and S. Dustdar, "A vector space search engine for Web services," presented at Third IEEE European Conference on Web Services, Sweden, 2005.
- [20]. N. Kokash, W.-J. v. d. Heuvel, and D. A., Vincenzo, "Leveraging Web Services Discovery with Customizable Hybrid Matching," Technical Report, University of Trento, vol. DIT-06-042, 2006.
- [21]. N. Kokash, "A Comparison of Web Service Interface Similarity Measures," University of Trento 2006.
- [22]. E. Stroulia and Y. Wang, "Structural and Semantic Matching for Accessing Web Service Similarity," *International Journal of Cooperation Information Systems*, vol. 14, pp. 407 - 437, 2005.
- [23]. J. Wu and Z. Wu, "Similarity-based Web Service Matchmaking," presented at IEEE International Conference on Service Computing, 2005.
- [24]. L. Lovasz and M. D. Plummer, *Matching Theory*. North-Holland: Elsevier Science Publisher, 1986.
- [25]. Zaremski and J. Wing, "Signature Matching of Software Components," *ACM Transactions on Software Engineering and Methodology*, pp. 333-369, 1997.
- [26]. Z. Zhuang, P. Mitra, and A. Jaiswal, "Corpus-based web services matchmaking," presented at Workshop on Exploring Planning and Scheduling for Web services, Grid, and Autonomic Computing, 2005.
- [27]. Richi Nayak Bryan Lee, "Web Service Discovery with additional Semantics and Clustering" 2007 IEEE/WIC/ACM International Conference on Web Intelligence, 2007
- [28]. W. Abramowicz, K. Haniewicz, M. Kaczmarek and D. Zyskowski "Architecture for Web services filtering and clustering. In *Internet and Web Applications and Services*", (ICIW '07), 2007.
- [29]. X. Dong, A. Halevy, J. Madhavan, E. Nemes and J. Zhang. Similarity Search for Web services. In *Proceedings of the 30th VLDB Conference*, Toronto, Canada, 2004.
- [30]. Constantinescu, W. Binder and B. Faltings. Flexible and efficient matchmaking and ranking in service directories. In *Proceedings of the IEEE International Conference on Web Services (ICWS'05)*, 2005
- [31]. Jiangang Ma, Yanchun Zhang, Jing He "Efficiently Finding Web Services Using a Clustering Semantic Approach" CSSSIA 2008, April 22, ACM ISBN 978-1-60558-107-1/08/04, 2008
- [32]. Wei Liu and Wilson Wong "Discovering Homogenous Service Communities through Web Service Clustering" SOCASE 2008, LNCS 5006, pp. 69-82, 2008. Springer-Verlag Berlin Heidelberg 2008
- [33]. "Discovering homogenous web service community in the user centric web environment", IEEE transition of service computing VOL.2 No@ April June 2009
- [34]. "Similarity search fo web service" Porceeding of the 30th VLDB conference, Totronto Canda 2004

Author Biographies

Augustine.K.V received his Masters degree in Mathematics in the year 1996 Later on in 2005 completed his Masters in Computer Application. He is doing his Masters of Technology in Computer Science and Engineering. He has also completed Post Graduate diploma in computer science, Statistical and Research Methodology. Currently he is working for the Government of Puducherry as statistician His research interest are computational statistics, Service Computing, data mining and Discrete Mathematics.

© Sprinter Global Publication, 2010

www.sprinterglobal.org/ijcset