Proceeding of International Conference on Computer Science and Information Technology ICCSIT-2012

Dr. Srikanta Patnaik

Follow this and additional works at: https://www.interscience.in/conf_proc_volumes

Part of the Computer Engineering Commons
Editorial

Thinking about the future can help you shape it. Thinking too much about the future can make you spacey. Our country is now in the IT era. More than 3.5 million people in India are employed in IT sector. This itself will explain you the significance of the IT sector in our country. IT sector is having a marvelous growth and a spectacular significance in India over the past decade. India's economy has also grown as well. Indian IT sector has placed its own in the third position among the world countries because of our countries domestic international market demand. In the last one decade, the growth of IT industry in India has shown a tremendous rate of 35%. India have reached a nearer GDP contribution of around 8.5% which is more or less equal to United States progress. IT sectors impact over India started because of the Y2Y problem that occurred before 2 decades, which in follow wanted a large number of skilled and efficient human resources to satisfy the database correction needs owing to the cope up with the later new adventures. In the past decade, India has become the significant destination among the Asian countries. IT growth in India is mainly because of the prominent role played by the following software series system namely: Custom Application Development and Maintenance (CADM), System Integration, IT Consulting, Application Management, Infrastructure Management Services, Software testing, Service-oriented architecture, Web services.

Let me highlight some of the recent developments in Computer Science & Engineering. Recently, many U.S. ISP have fallen in line with their international counterparts in capping monthly residential broadband usage. A new research by a Georgia Tech researcher shows such pricing models trigger uneasy user experiences that could be mitigated by better tools to monitor data usage through their home networks. The study found, typically manage their capped broadband access against three uncertainties invisible balances, mysterious processes and multiple users and these uncertainties have predictable impacts on household Internet use and can force difficult choices on users. Given the undeniable trend in both Internet norms (such as cloud-based applications) and home-entertainment delivery toward greater broadband requirements, the study seeks to create awareness and empathy among designers and researchers about the experience of Internet use under bandwidth caps. Now researchers at Northwestern University’s McCormick School of Engineering and Applied Science have discovered a way to ease that tension. Ono, a unique software solution that allows users to efficiently identify nearby P2P clients. The software, which is freely available and has been downloaded by more than 150,000 users, benefits ISPs by reducing costly cross-network traffic without sacrificing performance for the user. In fact, when ISPs configure their networks properly, their software significantly improves transfer speeds by as much as 207 percent on average.

The purpose of holding such conferences is to bring together, on a common platform, scientists, engineers and researchers along with other eminent academic personalities in electrical & electronics engineering. They would get ample scope to make exchange of views, ideas and thoughts besides presentation of papers based on intensive research in the areas of the conference theme and sub-topics. It’s my pleasure to welcome all the participants, delegates and organizer to this international conference on behalf of IOAJ family members. I sincerely thank all the authors for their invaluable contribution to this conference. I am indebted towards the reviewers and Board of Editors for their generous gifts of time, energy and effort.

Editor-in-Chief
Dr. Srikanta Patnaik
Chairman, I.I.M.T., Bhubaneswar
Intersceince Campus,
At/Po.: Kantabada, Via-Janla, Dist-Khurda
Bhubaneswar, Pin:752054. Orissa, INDIA.
Cloud Computing: A Holistic Approach To Cloud Security

Tarun Sinha, Vikas Lamba, Gulshan Jaiswal, Sumit Roy,
Madhvendra Misra, Rahul Gupta & Shashikant Rai
Indian Institute of Information Technology, Allahabad, India
E-mail: taruncs2004@gmail.com, vikas.lamba@hotmail.com, gulshan.2439139@gmail.com,
sumit.rr@gmail.com, drmadhvendra@gmail.com, rahul.gupta@iiita.ac.in, Shashikant@iiita.ac.in

Abstract – Data Protection has been concern for majority of the organization. Pertaining to the various researches that has been published regarding the security issue in the cloud architecture, till now there has not been any discussion regarding the role of Cloud Security Manager in any of them.

Cloud Security Manager or CSM being the pivot of this whole process is an important point in this Cloud Architecture. His role must be clearly defined and communicated. Throughout this research paper our primary concern is regarding the practices that must be followed by the CSM and also the necessary guidelines that should be followed in order to maintain the security and authenticity of the client’s data.

In some of the papers there has been discussion regarding the level of trust between the cloud service provider and their clients regarding the security of the data. Such issues actually require a transparent process that should be followed by the Cloud Service Provider. So, in this paper we try to solve these issues by providing a framework for Cloud Security Manager, Cloud Authentication Framework and also described the scope of a Cloud Security Manager.

These frameworks and scope for the CSM would actually help to increase the level of trust between the Cloud Service Provider and their clients. We try to cover all of the aspect related to CSM, however there could be some improvement regarding their job execution depending upon the continuously changing trend in the cloud industry.

Keywords - Cloud Security Manager, Cloud Authentication Framework

I. INTRODUCTION

Today, where more and more companies are moving towards Cloud Services to reduce costs and to keep a pace with latest technologies, the issue of cloud Security is more serious than ever. But before moving to Cloud security we should first understand what is Cloud Computing and why is it in so much demand these days. According to the definition of National Institute of Standards and Technology, U.S. Department of Commerce[1]

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a ……………..released with minimal management effort or service provider interaction.”

It is important that the IT services must provide valuable and economic benefit to the organization. For the sake of valuable contribution to the business, Cloud computing has provided a valuable input. However, use of cloud computing also raises the issue of security too.

Cloud computing provides better flexibility to the organization. This paper in response to the security concern related to cloud computing discusses about the role of cloud security manager, and also provides a framework for authentication. The entities associated with the cloud security and related issues are also discussed in view of current modern scenario.

II. CURRENT ISSUES WITH CLOUD COMPUTING

The Main issues attached with the Cloud Computing are

1. Privacy
2. Compliance
3. Security
4. Sustainability
5. Availability
6. Third-party data control
7. Vendor Lock-in or Data lock-in
With the increasing usage and complexity of the cloud, the issues associated with it are getting severe. Privacy of the client and the organizational data are being considered very important to be secured because of the value attached with it. Compliance and standardization of the processes used in the cloud along with security of the data, processes and other information are a matter of big concern for cloud industry. Further the challenge extends when privacy, compliance and security are not the only issues to be dealt with, but, sustainability and the availability of the services have to be dealt in par with other entities. Though a cloud security provider and associated client are obvious entities in a cloud, the presence of third party data control adds complexity to the situation which has to be managed with a holistic and balanced approach.

Ultimately, another new problem in the cloud horizon is the problem of the data lock-in or vendor lock-in which is quite hefty to deal if occurs. Although most of the clients are not aware of this particular situation, the vendor lock-in problem has a serious nomenclature and has to be handled carefully.

III. LITERATURE REVIEW

In the paper “Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control”[2] authors Richard Chow, Philippe Golle, Markus Jakobsson, Ryusuke Masuoka, Jesus Molina, Elaine Shi, & Jessica Staddon recognized some more problems like Authentication and Authorization.

In the paper “Towards Analyzing Data Security Risks in Cloud Computing Environments”[3] authors Amit Sangroya, Saurabh Kumar, Jaideep Dhok, and Vasudeva Varma suggested that trust is one of the key factors between Cloud Service provider and the clients. The lack of trust also increases the security concern.

In the paper “Current state of play: records management and the cloud”[4] authors Katharine Stuart and David Bromage found the incorporation of “the cloud” into the way organisations conduct business, should not be based on a technological decision, but should be based on a decision examining risk to an organization’s information.

IV. RELATED WORK

In the paper “Securing Cloud System via Internal Control Management”[5] authors Ashwin Alfred Pinto, Shvetank Verma, Satyam Singh, Prashant Srivastava, Rahul Gupta, and Vijay Kumar Chaurasiya tried to solve the issue of cloud security through a new entity called Cloud Security Manager. But they also found some problem in authentication of cloud security manager as he is the one who is going to access the cloud facilities made available to client. Also there is an absence of a framework for the scope of cloud security manager so we tried to give a framework for that and also a framework proposed for the authentication of CSM (Cloud Security Manager).

V. CSM and IT’S SCOPE

According to the introducers of Cloud security manager, He will be keeping records of each and every access done by clients as well dealing with the third party processing. But this will not be much for ensuring the Security and better management of Cloud Services for the client.

The CSM should be able to perform following tasks –

![Fig. 1: Tasks of CSM](image)

VI. PROPOSED FRAMEWORK

The Entities in the framework are:

1. CSP (Cloud Security provider): is the organisation which is providing the cloud services to one or many clients.
2. CSM (Cloud Security manager): is the one who is responsible for management and security of the cloud services.
3. Client/Customer: is the person or organisation who is subscribing the cloud services from the CSP.
Fig. 2 : Framework For Cloud Security Manager

The scope of CSM can be divided into 7 stages -

1. **Coordination**: The CSM will work as a medium between Client and Cloud service provider. He will ensure that CSP and Client both get benefits from this alliance. The Client gets the good and secure service as well as the CSP also gets the right price. CSM will also be responsible for understanding the needs of the client and get a best possible service for him from the service portfolio of CSP. He will be responsible for maintaining coordination with CSP, Client and Network Administrator for smooth working of system.

2. **Compliances**: The CSM will be responsible for taking the necessary steps to meet the compliances and laws made by government. He should plan the services for Client accordingly. He should also be able to find and remove any security issues in the plan and implementation of Services.

3. **Control Management**: The CSM should ensure that all change and configuration request should be accessed and completed in minimum time or according to the SLA. CSM should also be able to provide new services to the Client on his request in live environment whenever possible.

4. **Data Backup and recovery**: The CSM should also ensure the Backup of Client’s data for and unexpected failure of system. He should also be able to restore the services in minimum time.

5. **Incidence handling**: Any Security issue found during the service time should be documented and rectified. And the Services must be restored and recovered from any security incident.

6. **Reporting**: The CSM should record all the transactions made by client and should produce a report with the entire incidents having security and compliance issues. The report should be send to the Client and his own team from CSP.

7. **Enhancing Competencies and competitiveness**: The CSM should learn from previous incidents and history to increase the competencies of CSP and to provide maximum security to client. He should also able to acquire the best practices from different organisation and should use for its client. The Standards and norms followed by his organisation (CSP) should incorporate the best from the client’s previous Standard and norms. So that he can develop a better approach by using an enhanced Set of practices to secure the services.

The next and other important problem was since the CSM has so much access to the Clients data, how we can assure that only CSM has the access and other not, for which we developed a framework for authentication of CSM.

The Security and Access control policy can be set according to the Client’s requirement. Since the providing more secure and trusted authentication solution may increase the cost. Hence the Client should categories its data security level with the proposed Authentication Matrix:

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Minimum Level Required</th>
<th>Authentication System</th>
<th>Clients Data Type</th>
<th>May change according to client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td>Public</td>
<td></td>
<td>Public data</td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Private/ Hybrid/ Public</td>
<td>User ID and Password</td>
<td>Historical Data</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>Private/ Hybrid/ Public</td>
<td>Authentication with controlled policies for ID and Password</td>
<td>Daily transactions of files and user data</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>Private/ Hybrid/ Public</td>
<td>Multi-factor Authentication</td>
<td>Financial transactions</td>
<td></td>
</tr>
</tbody>
</table>

The Client and CSM should use the Authentication accordingly. The Security of data can be enhanced by using the following proposed framework. The Security level is defined as e authentication framework of Australia.

Layers in Authentication framework:

1. **User Layer**
2. **Cloud Authentication Layer**
3. **Cloud Service Layer**
Entities at different Layers

User layer:
- CSM
- Client

Cloud Authentication Layer:
- Authentication Server
- Policy Server

Cloud Service Layer:
- Client Database
- Web Server
- Database Server
- Application Server etc.

VII. CLOUD AUTHENTICATION FRAMEWORK THROUGH CSM, A LAYERED APPROACH

The Cloud Authentication Framework is primarily composed of three layers, namely, the User layer, the Cloud Authentication layer and the Cloud Service layer. Each layer has its own entities that are associated to other entities to serve the core purpose of the authentication process. The layered approach is chosen to show the process because it is easier to understand the process as it brings transparency and can be easily understood. The layers are mentioned in detail below:

1. **User Layer**: The User layer is comprised of two core entities: the Cloud Security Manager (CSM) and the Client. The interaction between these two entities is very significant as the Client tries to establish the trust in the Cloud service provider for the high end security of his data and information. The CSM on the other hand ensures the security of the data both at rest (storage), as well as at transit (transmission). The CSM is primarily responsible for allocating desired resources to the client for the data storage at the cloud. The Client’s data security status report is generated after a specific amount of time and is accessed both by the CSM and the Client so that proper trust be maintained between the two at every point of time. The CSM is also responsible for providing extra security and more communication for Key Clients (A key client is referred to the client who in contributing to a major chunk of the business). The Client maintains its existing security policy which is shown in the diagram as Client’s Existing Security policy (CESP). The responsibility of the CSM further enhances to incorporate the CESP with the cloud’s security. Both the CSM and the Client clearly understand the legal issues involved and maintain their consensus. The performance Objectives very important to both the CSM and the client and the CSM reviews the performance objectives and associated measures at quick intervals.

2. **Cloud Authentication Layer**: The Cloud authentication layer is in perfect coordination with the User layer. This layer contains the Authentication server which contains various encryption techniques and the policy server which is primarily concerned with the level of the security as mentioned earlier i.e. No authentication, Low, Moderate, High and Extreme level Security. The Cloud authentication layer, thus, serves as the intermediate interface between the User layer and the Cloud Service layer.

3. **Cloud Service Layer**: The Cloud Service Manager can access various services of the Cloud service layer after the authentication is done under specific policy framework. The cloud service layer is consisted of the Client metadata, the database

Fig.3: Cloud Authentication Framework
server, web servers and the Application servers. The CSM again is responsible for proper separation of one Client’s data with that of another by using standard mechanism such as using Virtual Machines. The Client on the other hand should be enquiring about the location of his data and proper management of the data and associated applications.

**VIII. WORKING OF AUTHENTICATION FRAMEWORK**

![Flowchart of Authentication](image)

Working of this Cloud Authentication Framework is basically based on the requirement of the clients. As shown in the matrix this authentication framework is based on the 3 categorized needs of the clients, i.e. Low, Medium and High. Based on these requirements as classified by the clients, the minimum requirement of the cloud type has to be met by the cloud service provider, so as to maintain the minimum level of security in the cloud architecture. These minimum requirements are the public, hybrid and private network.

In the above shown Flowchart, Users and CSM have to pass their request to Authentication Server where it checks whether the User or CSM is authorized or not and if authorized up to what level he has permission to access data. For this the policies regarding access control should be implemented on policy server according to the Client’s directions.

Policy server will use Chinese wall method to separate the user policies and data with each other on different servers.

According to the policies of access control the data can be accessed by the CSM and according to that only he prepares the transaction report for and if he finds any anomaly in the transaction or access he can report to his team for security breach and ask them to rectify the problem.

Then for the one month he consolidates all the security issues and breaches happened in cloud and send it to the Client for his knowledge.

**IX. CONCLUSION**

The CSM is very novel concept and need to be developed further we tried our best to make a framework for his Scope and nature of work. We also tried to provide an authentication framework for the CSM so that Client can be assured of maximum security.

The Cloud Security manager can be the person who can bring trust between the Client and the CSP. He can also help the organization to enhance the security and help the CSP to improve the Security standards.

**X. FUTURE WORK**

Although we tried our best to accommodate all possible work and Scope of CSM but it is always possible to introduce new and better scope of work for this. Similarly Authentication Framework is also can be tweaked according to the Client’s need.

We have not implemented this framework in any business organization so the implementation and checking its practicality is one of our future works.

**REFERENCES AND CITATIONS**


Structured and Ordered XML Tree Pattern Matching Algorithms

M. Rajani Kanth & D. N. V. Rajesh K
Dept of Computer Science and Engineering, Sri Sunflower College of Engineering and Technology, Lankapalli 531131
E-mail: rajinikanthmeka7@yahoo.co.in, rajsri.be@gmail.com

Abstract – As business and enterprises generate and exchange XML data more often, there is an increasing need for efficient processing of queries on XML data. Searching for the occurrences of a tree pattern query in an XML database is a core operation in XML query processing. Prior works demonstrate that holistic twig pattern matching algorithm is an efficient technique to answer an XML tree pattern with parent-child (P-C) and ancestor-descendant (A-D) relationships, as it can effectively control the size of intermediate results during query processing. However, XML query languages (e.g. XPath, XQuery) define more axes and functions such as negation function, order-based axis and wildcards. In this article, we research a large set of XML tree pattern, called extended XML tree pattern, which may include P-C, A-D relationships, negation functions, wildcards and order restriction. We establish a theoretical framework about “matching cross” which demonstrates the intrinsic reason in the proof of optimality on holistic algorithms. Based on our theorems, we propose a set of novel algorithms to efficiently process three categories of extended XML tree patterns. A set of experimental results on both real-life and synthetic data sets demonstrate the effectiveness and efficiency of our proposed theories and algorithms.

I. INTRODUCTION

As business and enterprises generate and exchange XML data more often, there is an increasing need for efficient processing of queries on XML data. An XML query pattern commonly can be represented as a rooted, labeled tree (or called twig). For example, Figure 1(a) shows an example XPath query: A[B]=C and the corresponding XML tree pattern. This query finds all node C that has the parent A which has another child B. In Figure (b), the query answers are nodes “C1” and “C2”.

Fig. 1: Example XML tree query and document. “ ” denotes the return node in query. The answers are C1 and C2. The digital labels will be explained later. Efficient matching of XML tree patterns has been widely considered as a core operation in XML query processing. In recent years, many methods have been proposed to match XML tree queries efficiently. In Extended XML tree pattern Previous algorithms focus on XML tree pattern queries with only P-C and A-D relationships. Little work has been done on XML tree queries which may contain wildcards, negation function and order restriction, all of which are frequently used in XML query languages such as XPath and XQuery. In this article, we call an XML tree pattern with negation function, wildcards and/or order restriction as extended XML tree pattern. Figure 2, for example, shows four extended XML tree patterns. Query includes a wildcard node “*”, which can match any single node in an XML database. Query includes a negative edge, denoted by ‘~’. This query finds A that has a child B, but has no child C. In XPath language, the semantic of negative edge can be presented with “not” boolean function. Query has the order restriction, which is equivalent to an XPath “/A/B[following-sibling::C]”. The ‘<’ in a box shows that all children under A are ordered. The semantics of order-base tree pattern is captured by a mapping from the pattern nodes to nodes in an XML database such that the structural and ordered relationships are satisfied. Finally, Query is more complicated, which contains wildcards, negation function and order restriction.
II. RELATED WORK

In the context of semi-structured and XML databases, tree based query pattern is a very practical and important class of queries. Lore DBMS and Timber systems have considered various aspects of query processing on such data and queries. XML data and various issues in their storage as well as query processing using relational database systems have recently been considered. Our holistic algorithm TreeMatch for extended tree patterns can leverage these previous techniques. From the aspect of theoretical research about the optimality of XML tree pattern matching, Choi et al. developed theorems to prove that it is impossible to devise a holistic algorithm to guarantee the optimality for queries with any combination of P-C and A-D relationships. Shalem et al. Research the space complexity of processing XML twig queries. Their paper showed that the upper bound of full-fledged queries with parent-child and ancestor-descendant edges are $O(D)$, where $D$ is the document size. In other words, their results also theoretically prove that there exists no algorithm to optimally process an arbitrary query $Q = \ldots$. Our research in this article moves the frontier forward by identifying a large subclass of $Q = \ldots$, which can be guaranteed to process optimally.

2.2.1 The algorithm in this paper:

TreeMatch for $Q/\ldots/\ast$

There is an input list $T_q$ associated with each query node $q$, in which all the elements have the same tag name $q$. Thus, we use $eq$ to refer to these elements. $cur(T_q)$ denotes the current element pointed by the cursor of $T_q$. The cursor can be advanced to the next element in $T_q$ with the procedure $advance(T_q)$.

Algorithm 1: Algorithm TreeMatch for class $Q/\ldots/\ast$

```
1: locateMatchLabel(Q);
2: while (end(root node)) do
3:   next = getNext(topBranchingNode);
4:   if (next is a return node)
5:     addToList(NAB(next), cur(T_next));
6:   advance(T_next); // read the next element in $T_{\text{match}}$
7:   updateGet(T_next, $\text{encoding}$;
8:   locateMatchLabel(Q); // locate next element with matching path
9:   emptyAllSets(root);
```

TreeMatch

Now we go through Algorithm 1. Line 1 locates the first elements whose paths match the individual root-leaf path pattern. In each iteration, a leaf node $fact$ is selected by $get\text{Next}$ function (line 3). The purpose of line 4, 5 is to insert the potential matching elements to $outputlist$. Line 6 advances the list $T_{\text{fact}}$ and line 7 updates the set encoding. Line 8 locates the next matching element to the individual path. Finally, when all data have been processed, we need to empty all sets in Procedure $Empty\text{AllSets}$ (Line 9) to guarantee the completeness of output solutions.

Algorithm 2: Procedures and Functions in TreeMatch

```
1: Procedure locateMatchLabel(Q):
2:   for each leaf $q \in Q$, locate the extended Dewey label $eq$ in $T_q$ such that $eq$ matches the individual root-leaf path
3: Procedure addToOutputList(q, $eq$):
4:   for each $eq \in S_q$ do
5:     if (satisfyTreePattern($eq$, $eq$))
6:       outputList(c) = add($eq$);
7: Function satisfyTreePattern($eq$, $eq$):
8:   if (bitVector($eq$, $eq$) = 1) return true;
9:   else return false;
10: Procedure updateSet(q, $eq$):
11:   cleanSet(q, $eq$);
12:   add $e$ to set $S_q$; // set the proper bitVector($e$)
13:   if (isValidBranching(q))
14:     outputSet(c) = add(outputList(c), $eq$);
15: else if (isLeaf(q) && (bitVector($e$) = 1..))
16:     updateAncestorSet(q);
17: Procedure cleanSet(q, $eq$):
18:   for each element $e \in S_q$ do
19:     if (isAtemeTreePattern($eq$, $eq$))
20:       if ($q$ is a return node)
21:         addToList(NAB($eq$), $eq$);
22: else if (isValidBranching(q))
23:     if (isLeaf(q) && (bitVector($e$) = 1..))
24:       updateAncestorSet($eq$);
25: Procedure emptyAllSets(q):
26:   if ($q$ is not a leaf node)
27:     for each child $e$ of $q$ do $Empty\text{AllSets}(e)$;
28:   if ($q$ is a leaf node)
29:     for each child $e$ of $q$ do $Empty\text{AllSets}(e)$;
```

Algorithm $get\text{Next}$ (see Algorithm 2) is the core function called in TreeMatch, in which we accomplish two tasks. For the first task to identify the next processed node, Algorithm $get\text{Next}(n)$ returns a query leaf node $f$ according to the following recursive criteria (i) if $n$ is a leaf node, $f = n$ (Line 2); else (ii) $n$ is a branching node, then suppose element $e$ matches node
III. PROPOSED WORK:

We present an extensive experimental study of TreeMatch on real-life and synthetic data sets. Our results verify the effectiveness, in terms of accuracy and optimality, of the TreeMatch as holistic twig join algorithms for large XML data sets. These benefits become apparent in a comparison to previously four proposed algorithms. The reason that we choose these algorithms for comparison is that (1) similar to TreeMatch, both TJFast and TwigStack are two holistic twig pattern matching algorithms. But they cannot process queries with order restriction or negative edges; and (2) OrderedTJ is a holistic twig algorithm which can handle order-based XML tree pattern, but is not appropriate for queries with negative edges; and finally (3) TwigStackListNot is proposed for queries with negative edges, but it cannot work for ordered queries. Only TreeMatch algorithm can process queries with order restriction, negative edge and wildcards.

Experiment Settings and Dataset

We implemented all tested algorithms in JDK 1.4 using the file system as a simple storage engine. We conducted all the experiments on a computer with Intel Pentium IV 1.7GHz CPU and 2G of RAM. To offer a comprehensive evaluation of our new algorithms, we conducted experiments on both synthetic and real XML data. The synthetic dataset is generated randomly. There are totally 7 tags $A, B, \ldots, F, G$ in the dataset and tags are assigned uniformly from them.

The real data are DBLP (highly regular) and Treebank (highly irregular), which are included to test the two extremes of the spectrum in terms of the structural complexity. The recursive structure in TreeBank is deep (average depth: 7.8, maximal depth: 36). We can easily find queries on this dataset to demonstrate the sub-optimality for our tested algorithms.

IV. EXPERIMENTAL RESULTS
V. CONCLUSION AND FUTURE WORK

We have introduced a notion of matching cross to address the problem of the sub-optimality in holistic XML tree pattern matching algorithms. We have identified a large optimal query classes for three kinds of queries, that is $Q=;==;\Downarrow$, $Q=;==;\Downarrow;<$ and $Q=;==;\Downarrow;<;:$. Based on these results, we have proposed a new holistic algorithm called TreeMatch to achieve such theoretical optimal query classes. Finally, extensive experiments demonstrate the advantage of our algorithms and verify the correctness of theoretical results.

REFERENCES


HTML Parser and Support Vector Machine used for Automatic Template Extraction

C. SreeDeepak & S. Kusuma
Dept. of CSE, Madanapalli Institute of Technology and Science, Madanapalli, Andhra Pradesh, India

Abstract – Extracting data from Web pages using wrappers is a fundamental problem arising in a large variety of applications of vast practical interests. There are two main issues relevant to Web-data extraction, namely wrapper generation and wrapper maintenance. In this paper, we propose a novel schema-guided approach to the problem of automatic wrapper maintenance. It is based on the observation that despite various page changes, many important features of the pages are preserved, such as syntactic patterns, annotations, and hyperlinks of the extracted data items. Our approach uses these preserved features to identify the locations of the desired values in the changed pages, and repair wrappers.

In this paper, we present novel algorithms for extracting templates from a large number of web documents which are generated from heterogeneous templates. We cluster the web documents based on the similarity of underlying template structures in the documents so that the template for each cluster is extracted simultaneously. We develop a novel goodness measure with its fast approximation for clustering and provide comprehensive analysis of our algorithm. Our experimental results with real-life data sets confirm the effectiveness and robustness of our algorithm compared to the state of the art for template detection algorithms.

I. INTRODUCTION

World Wide Web is the most useful source of information. In order to achieve high productivity of publishing, the WebPages in many websites are automatically populated by using the common templates with contents. The templates provide readers easy access to the contents guided by consistent structures. However, for machines, the templates are considered harmful since they degrade the accuracy and performance of web applications due to irrelevant terms in templates. Thus, template detection techniques have received a lot of attention recently to improve the performance of search engines, clustering, and classification of web documents.

The Web poses itself as the largest data repository ever available in the history of humankind. Major efforts have been made in order to provide efficient access to relevant information within this huge repository of data. Although several techniques have been developed to the problem of Web data extraction, their use is still not spread, mostly because of the need for high human intervention and the low quality of the extraction results. In this paper, we present a domain-oriented approach to Web data extraction and discuss its application to automatically extracting news from Web sites. Our approach is based on a highly efficient tree structure analysis that produces very effective results. We have tested our approach with several important Brazilian on-line news sites and achieved very precise results, correctly extracting 87.71% of the news in a set of 4088 pages distributed among 35 different sites.

In this paper, we present novel algorithms for extracting templates from a large number of web documents which are generated from heterogeneous templates. We cluster the web documents based on the similarity of underlying template structures in the documents so that the template for each cluster is extracted simultaneously. We develop a novel goodness measure with its fast approximation for clustering and provide comprehensive analysis of our algorithm. Our experimental results with real-life data sets confirm the effectiveness and robustness of our algorithm compared to the state of the art for template detection algorithms.

II. RELATED WORK:

Recently, Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. In this phase, various modeling techniques are
selected and applied, and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase is often needed.

The wide-spread use of distributed information systems leads to the construction of large data collections in business, science and on the Web. These data collections contain a wealth of information, that however needs to be discovered. Businesses can learn from their transaction data more about the behavior of their customers and therefore can improve their business by exploiting this knowledge. Science can obtain from observational data (e.g. satellite data) new insights on research questions. Web usage information can be analyzed and exploited to optimize information access. Data mining provides methods that allow to extract from large data collections unknown relationships among the data items that are useful for decision making. Thus data mining generates novel, unsuspected interpretations of data.

III. A NOVEL APPROACH OF PROPOSED SYSTEM:

Newcomers to Perl often want to know how to parse HTML. For instance, to extract the text between <p> and </p> tags, or to extract content by assembling and following hyperlinks. HTML is treacherous in that in looks as though it could be handled with just a few regular expressions. Even when you slurp the whole file and work on large strings, sooner or later regular expressions won't be enough. The HTML::Parser module provides powerful mechanisms for extracting content, tags and tag attributes from any html stream.

A support vector machine (SVM) is a concept in statistics and computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes comprises the input, making the SVM a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

IV. HYPOTHESIS FORMULATION AND VARIABLE SELECTION:

To keep the computational load reasonable, the mapping used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $K(x,y)$ selected to suit the problem. The hyperplanes in the higher dimensional space are defined as the set of points whose inner product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters $\alpha_i$ of images of feature vectors that occur in the data base. With this choice of a hyperplane, the points $x$ in the feature space that are mapped into the hyperplane are defined by the relation:

$$\sum_i \alpha_i K(x_i, x) = \text{constant}$$

Note that if $K(x,y)$ becomes small as $y$ grows further from $x$, each element in the sum measures the degree of closeness of the test point $x$ to the corresponding data base point $x_i$. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points $x$ mapped into any hyperplane can be quite convoluted as a result allowing much more complex discrimination between sets which are not convex at all in the original space.
H3 (green) doesn’t separate the two classes. H1 (blue) does, with a small margin and H2 (red) with the maximum margin.

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as a $p$-dimensional vector (a list of $p$ numbers), and we want to know whether we can separate such points with a $(p - 1)$-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier; or equivalently, the perceptron of optimal stability.

V. RESULT ANALYSIS:

The screen shots of the paper is displayed below pictures.

1) The front end of the page.

2)  

3)  

4)  

VI. CONCLUSION:

We introduced HTML parser for extracting the template from the input data. Extracting and maintaining the data from the web pages is a big problem, so we used the parser for the easy extraction. We introduced a novel approach of the template detection from heterogeneous web documents. We employed the Support vector machine method a non-probabilistic linear classifier, its both the clustering and classification technique that improves the efficiency in extracting the common template.
REFERENCES:


[2] Heterogeneous Web Data Extraction using Ontology Hicham Snoussi, Laurent Magnin, Jian-Yun Nie


[4] Experiences with Content Extraction from the Web Mira Dontcheva, Sharon Lin, Steven M. Drucker, David Salesin, Michael F. Cohen

[5] Information Extraction from Heterogeneous Sources Using Domain Ontologies Waqas-ur-Rehman Chaudhry, Farid Meziane


[7] Automatic Web Data Extraction based on Genetic Algorithms and Regular Expressions David F. Barrero and David Camacho and Maria D. Moreno

❖❖❖
Android Application for Measuring Human Body Temperature
And Forecasting Future Illness

S Prabhakaran¹, Dhaneshwari Kumari² & Ria Ahuja³
¹Dept. of IT, ²³Dept. of CSE,
¹,²³Saveetha School of Engineering, Saveetha University, Chennai, India
E-mail : prof.prabhakar.sse@gmail.com¹, meetdhani18@gmail.com², ria.ahuja10@gmail.com³

Abstract – Android Application for measuring human body temperature is a new age mobile thermometer. This kind of application already exists but requires manual feeding temperature. In our project, we propose an application which will measure the body temperature automatically while the user is operating the mobile device. It has an in-built function which can trigger alert messages whenever the temperature becomes critical more than normal human body temperature. The display segment of the device is made up of capacitive touch screen, which can act upon the bioelectricity produced by human body with each and every touch. This application requires Android Operating System Version 2.2. It will also diagnose the other diseases the user might have depending upon the symptoms entered.

Keywords - Android; capacitive touch screen; capacitance; bioelectricity; temperature;

I. INTRODUCTION
The human body temperature measuring application makes use of the very popular and versatile android technology. Android operating system is used as it is an open source platform where applications can easily be customized as per the user’s requirement. The capacitive touch screen of the android smartphones is sensitive to human touch. The bioelectricity flowing continuously through our body is responsible for the touch detection. This application makes use of the capacitive touch screen along with the human electricity to measure and display the body temperature with the probable illness. It has an internal database with information about the symptoms of various diseases and their associated temperature ranges. The main advantage of this application –

- It can act as a medical kit complete with thermometer.
- It is easy to install and use.
- You can carry it easily as you don’t have to worry about the fragility of a glass tube.

This application can be used for all Android versions starting from version 2.0 (Android Eclair).

This application also asks the user if he is experiencing any other symptoms and takes those in. Diseases along with their associated symptoms are stored in the database. Hence, based on the body temperature and the symptoms, it will display the disease that user might be experiencing.

II. CAPACITIVE TOUCH SCREEN

Capacitive touch screen are popularly used in electronic devices such as cellular phones, laptop touchpad etc., making use of the touch technology.

Fig. 1 : Capacitive Touch Screen
These touch screens are sensitive to human touch or any other conducting material. Capacitive sensing makes use of the bioelectricity [8] [9] (continuous electric flow through the body) as a conducting medium along with a steady electric current already flowing through the screens’ lower layer.

Capacitive touch screens [2] come in varieties such as Surface capacitance, Mutual-Capacitance, Self-Capacitance. Multi-touch and Gesture-Based technologies also make use capacitive sensing [1] [5]. Android phones make use of the self-capacitance touch screen.

In our application, we make use of the self-capacitance touch screen where the human finger acts as a conductor along with the conductor present in the device, thus, completing the circuit.

III. LAYERS OF CAPACITIVE TOUCH SCREEN

The capacitive touch screen [2] is made up of the following layers such as Substrate, Set of Electrodes, Panel made of nonconductive material as glass.

The substrate acts as a base for carry the electrodes. The substrates should be nonconductive and should not retain water contained in the atmosphere. Materials such as glass are excellent for these purposes. A constant voltage is passed through the electrodes, usually ITO (Indium Tin Oxide). The two layers of electrodes glued together using a pressure sensitive adhesive (PSA) to reduce the air gap. The air gap works as an insulator and reduces the touch sensitivity. The layers form a matrix of rows and columns, the X-Y grid. The current is passed through the rows and columns independent of each other [7] [17].

When the user touches the screen, a voltage drop is created at the point of contact and the capacitance of the touch screen at that point decreases. The current from all the four corners rushes to that point and the capacitive touch sensor located the position of the touch with the help of the X-Y grid. This information is then sent to the micro controller which then takes the required action [5].

IV. SYSTEM ARCHITECTURE

In our paper, we use the property, capacitance increases with increase in temperature. The finger is first placed within the indicated area on the cell phone screen for a particular time period i.e. 10 milliseconds which is responsible for accumulation of charge on the screen. This creates a drop in capacitance of touch screen as constant voltage from the four corners of the screen rushes towards the point of contact. The X-Y co-ordinates of the contact point are detected by the capacitive touch sensor. The sensor then, relays the information to the microcontroller [5]. The capacitance drop is measured against the usual drop and the respected temperature is calculated and displayed onto the screen. The user is then asked whether he is experiencing any other symptoms besides high temperature. The symptoms are then matched with the information stored in the internal database. The disease associated with the entered symptoms is then displayed to the user.

V. DATA FLOW DIAGRAM

The user should touch the indicated area for about 10 milliseconds or till he hears a beep. If the contact is less than the indicated time or if the position of the finger is incorrect, the user will be asked to touch the screen again. The capacitance drop is measured by the capacitive touch sensor which relays this information to the microcontroller [5]. The temperature as per the capacitance drop is calculated and displayed onto the screen.
VI. INPUT SENSING

Fig. 4 : Input Sensing

A constant voltage continuously flows through the lower layer of the glass in a capacitive touch screen. This voltage also has a constant capacitance say 1pF. When the upper glass layer is touched, the electric field of that particular area is disturbed. Hence, there is a drop in capacitance at the point of contact which is detected by the capacitive touch sensor using the X-Y grid.

In general, when the temperature of the human body raises, its electrical activity increases [8] [9]. Hence, when someone with a fever touches the upper glass layer, the drop in the capacitance is relatively less as capacitance increases with increase in temperature. Thus, this decrease in capacitance drop is measured against the usual drop and the body temperature is displayed.

VII. CALCULATE TEMPERATURE

The changed capacitance is stored in a variable. It is compared with the already stored value taken for normal body temperature i.e. 37.5°C.
The temperature is calculated as per the formula.

\[ Q = I \times t \]
\[ C = Q / V \]
\[ \Delta C = C_0 - C_t \]

Where,
- \( Q \) = Charge Accumulation on screen
- \( I \) = Amount of bioelectricity
- \( t \) = Specified time
- \( V \) = Amount of applied voltage from screen corner
- \( \Delta C \) = Change in capacitance of screen
- \( C_0 \) = Untouched Capacitance
- \( C_t \) = Touched Capacitance

**VIII. DISPLAY TEMPERATURE AND DISEASE**

Once the capacitance drop is measured and the temperature is calculated, it is displayed onto the screen on a Fahrenheit (°F) scale.

It asks if there are any other symptoms besides fever that the user experiences and takes in those symptoms. It then searches its internal database for the possible illness the user might have [11] [12] [13] [14] [15] [16].

Table 1. Disease Table

<table>
<thead>
<tr>
<th>S. No.</th>
<th>DISEASE</th>
<th>TEMPERATURE</th>
<th>SYMPTOMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Viral Fever</td>
<td>38 - 39 °C (99.5 - 102.2 °F)</td>
<td>Body and muscle ache, redness of eyes, running nose, headache, cough</td>
</tr>
<tr>
<td>2</td>
<td>Dengue Fever</td>
<td>40° C (104° F)</td>
<td>Chills, headache, pain upon moving the eyes, and low backache.</td>
</tr>
<tr>
<td>3</td>
<td>Pneumonia</td>
<td>40° C (104° F)</td>
<td>Shaking chills, Sneezing, Sour Throat, Cough</td>
</tr>
</tbody>
</table>

Once it identifies the illness, it displays it to the user so he/she can take the necessary precautions and medication.

**IX. CONCLUSION**

The capacitive touch screen works on the principle of change in capacitance on contact with a conductor such as human finger. The human body has electric current flowing through it continuously which acts as a conductor for the capacitive sensing. The increase in temperature leads to increase in capacitance. Hence, we can conclude that rise in body temperature will lead to decrease in capacitance change and this change can be measured in order to calculate and display body temperature.

**X. FUTURE ENHANCEMENTS**

This App. can be enhanced further by having an alert system which regularly reminds the user to check his/her temperature. The user can store this information in the database. We can also include the functionality of sending an SMS to the users’ family doctor about the patients’ medical condition. A graph can also be plotted on a day-to-day basis.

**ACKNOWLEDGMENT**

We sincerely thank Dr. P. Shankar, Principal and Dr. R. Nedunchelian, H.O.D of Computer Science & Engineering Department of Saveetha School of Engineering for their support and encouragement to complete our work.

**REFERENCES**

[1] Touchscreen,  
http://en.wikipedia.org/wiki/Touchscreen


[15] Fever,


[19] Android Developers
Developer.android.com/guide/index.html

+++
A Proposed Approach for Storage Throughput Improvement in Android Phones

Shijin P.V.1, Rajesh M2 & Saugata Das Purkayastha3
1&2Embedded Systems, National Institute Of Electronics And IT, Calicut, India
3Smartphone & Tablet Solutions, ST Ericsson India Private Limited, Bangalore, India
E-mail : Shijin931@gmail.com1, rajesh@doeacccalicut.ac.in2, Saugata.das@stericsson.com3

Abstract – Today most smart phones use eMMC and SD card as storage devices. Storage performance improvement is one of the most important studies in the mobile phone industry today. Performance of many key mobile phone use cases like audio/video record/play, USB mass storage are linked to storage throughput. This paper investigated the cause of current storage performance bottleneck on the Android based smart phones and proposes an approach for storage throughput improvement by modifying the I/O architecture and device driver interface of the Linux kernel.

Keywords - Linux; ext4; eMMC; SD card; smart phone; Storage; Throughput; Block driver; I/O Scheduler; VFS; Disk cache

1. INTRODUCTION

Today’s fastest and most demanding embedded applications, such as multifunctional smart phones, PMP’s and tablet computers, require equally capable memory management solutions to provide a rich end user experience. Such devices most commonly employ flash devices for content storage.

Typically, flash memory is controlled by a dedicated controller that supervises data reads and writes, and operates under the control of the application’s CPU. However with developments in semiconductor technologies, memory densities are advancing at rapid rates, resulting in memory chips with ever -increasing capacities and making it difficult to keep the controller off the flash memory die. To maintain the required data rates and throughputs for high density chips designed to store high resolution video and to provide enhanced storage capabilities, the eMMC standard was developed. eMMC is a standard for embedded memory devices that contain not only a data storage element ( such as NAND flash memory), but also a controller for the storage element integrated on the same silicon die. The eMMC solution consists of three components – the multimedia card interface, the flash memory, and the flash memory controller in a standard BGA package.

Storage performance improvement is one of the most important studies in the mobile phone industry today. A sluggish mobile can never meet the increasing expectation of the end user. There are many ways to improve the storage throughput. An obvious solution is to improve the performance of the storage device.ie, using better flash storage or a faster non-volatile memory such as PCM. Indeed flash fabrication technology itself is improving at a fair pace; scaling trends projects flash to double in capacity every two years until the year 2016 [1]. However when it comes to performance, cost pressures in the consumer market are driving manufacturers to move away from the more reliable, higher performing SLC flash to the less reliable lower performing MLC or TLC flash; this makes it harder to rely solely on improvements due to flash scaling. Findings reveal that performance of a relatively small fraction of I/O traffic is responsible for a large fraction of overall application performance [2]. A more efficient solution is thus to use the faster storage media as a persistent write buffer for the performance sensitive I/O traffic. Even though very few papers are presented in storage throughput improvement, some relevant papers include vectored read : Exploiting the read performance of Hybrid NAND flash [3], A hardware filesystem implementation for high speed secondary storage[4], and User level techniques for improvement of disk i/o in WWW caching[5].

eMMC 4.5 is the upcoming embedded multimedia card standard and it is having a feature called context-ID .This feature can be used to improve the throughput of mobile phones and it necessitates source code modification in the block device operation path . This paper proposes an inode based context management for the popular ext4 filesystem and it also gives a
II. PRELIMINARIES

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android architecture consists of Applications, Application framework, Libraries, and the Linux kernel. Android relies on Linux for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.[7]

Each operation on a block device driver involves a large number of kernel components; the most important ones are shown in figure 1. For instance, a process issued a read system call on some disk file. The kernel typically does service as follows. [6]

![Virtual Filesystem Diagram]

Fig. 1 : Kernel components involved in Block device operation and the storage media (eMMC).

- The service routine of the read() system call activates a suitable VFS function, passing to it a file descriptor and an offset inside the file. The virtual file system is the upper layer of the block device handling architecture and it provides a common file model adopted by all filesystems supported by Linux.
- The VFS function determines if the requested data is already available and if necessary, how to perform the read operation. Sometimes there is no need to access the data on disk, because the kernel keeps in RAM the data most recently read from or written to – a block device.
- Assume that the kernel must read the data from the block device, thus it must determine the physical location of that data. To do this kernel relies on the mapping layer, which typically executes two steps.(a)It determines the block size of the filesystem including the file and computes the extent of the requested data in terms of file block numbers. Essentially the file is seen as split in many blocks, and the kernel determines the numbers (indices relative to the beginning of the file) of the blocks containing the requested data.(b)The mapping layer invokes a filesystem specific function that accesses the file’s disk inode and determines the position of the requested data on disk in terms of logical block numbers. Essentially the disk is seen as split in blocks, and the kernel determines the numbers corresponding to the blocks storing the requested data. Because a file may be stored in non adjacent blocks on disk, a data structure stored in the disk inode map each file block number to a logical block number.
- The kernel can now issue the read operation on the block device. It makes use of the generic block layer, which starts the I/O operations that transfer the requested data. In general each I/O operation involves a group of blocks that are adjacent on disk. Because the requested data is not necessarily adjacent on disk, the generic block layer might start several I/O operations. Each I/O operation is represented by a “block I/O” (in short bio) structure, which collects all information needed by the lower components to satisfy request.
- Below the generic block layer, the “I/O scheduler” sorts the pending I/O data transfer requests according to predefined kernel policies. The purpose of the scheduler is to group requests of data that lie near each other on the physical medium.
- Finally the block device drivers take care of the actual data transfer by sending suitable commands to the hardware interfaces of the disk controllers.

Presently requests are merged in the I/O scheduler based on its data direction and physical contiguity. For example if three applications A1, A2, A3 are writing data belonging to the file F1, F2, F3 continuously. Let f1, f2, f3 be the requests corresponding to the file F1, F2 and F3. In the driver level the requests of different files are mixed and merging will happen in I/O scheduler as stated above. In that case the queue may look like “f1 f2 f1 f2 f3 f1 f2 f3 f1 f3”. Theoretically these results in file fragmentation and this can slow down the throughput also.

Smart phone uses eMMC as its internal storage and the upcoming eMMC 4.5 supports a feature called
context-ID. To better differentiate between large sequential operations and small random operations, and to improve multitasking support, contexts can be associated with groups of read or write commands. Associating a group of commands with a single context allows the device to optimize handling of the data. A context can be seen as an active session, configured for a certain read/write pattern. Multiple read/write commands are associated with this context to create some logical association between them, to allow device to optimize performance. For example a large sequential write pattern may have better performance by allowing the device to improve internal locality and reduce some overheads. ex: If some large unit of data is allowed to be affected by a power failure as a whole while it is written to, all of the commands that fill this unit can work faster because they can reduce the overhead usually required to protect each write individually in case of power failure). Furthermore, handling of multiple concurrent contexts allows the device to reorganize each of the write patterns without getting confused when they are all mixed together.

Device may support one or more concurrent contexts, defined by a context-ID. Context-ID #0 always exists for backward compatibility and for context less data. Each context-ID (besides #0), has a configuration field in EXT_CSD register to control it’s behavior.

To use a context, an available context-ID shall be picked. Then it shall be initialized by writing the configuration register. Then data can be read/written associated to the context by specifying the context-ID in SET_BLOCK_COUNT command (CMD 23) before sending the read/write command. When the context is no longer needed, the configuration register should be updated to close the context-ID. A context shell be closed prior to re-configuring it for another configuration or use.

Before any contexts can be used it shall be configured, except for context #0 which is always preconfigured and cannot be changed. Configuration is done by writing to the configuration register of the specific context-ID required. Then all read commands or write commands that are associated with this context-ID shall be sent with ID. When the context is no longer needed, it should be closed by writing a zero byte to the configuration register.

The configuration registers are an array of 15 byte registers, one for each context (except the fixed predefined #0).

III. PROPOSAL IMPLEMENTATION

In this paper we are presenting a proposal to implement the context ID feature of the eMMC 4.5 based on the file inode number. We are planning to implement the proposal for the widespread ext4 filesystem. Enabling the card to exploit this feature necessiates a lot of exploration in the kernel components which are getting affected by a block device operation.

In Ext4 filesystem each file will be having a unique inode structure. Inode number “i_ino“ is a field element of this structure and this number can uniquely identify a file. At present there is no file identifier in the driver level and the I/O scheduler simply merges the requests based upon the data direction and physical contiguity. This results in merging of requests which do not belong to the same file. Eventually this leads to the file fragmentation.

![Fig. 2 : A buffer page including four buffers and their buffer heads.](image)

From the figure 2 it is clear that each of the requests sent by different applications are grouped in buffer heads and more than one buffer heads will be pointing to the buffers of same page. Based upon certain conditions these pages are flushed later. So each of the buffer head is converted in to a corresponding “Block I/O “(bio) structure and sent to the generic block layer. There would be a request queue in the block layer and the I/O scheduler will be merging these bios. Earlier the I/O scheduler will merge the requests based upon the data direction and physical contiguity alone. Now we need to avoid a cross inode merging by including one more comparison statement inside the function which is doing the merging. Also we need a flag in the bio structure which signals the file close for closing the context. Also there should be some flag to differentiate between normal data and metadata.

We created a structure in the struct buffer_head as below.

```c
struct_buffer_head {…
struct_context_id{
```

International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati
Unsigned long file_id;
Unsigned int file_close;
} cid;
...

We found that in the function submit_bh(); bh->b_page->mapping->host of struct buffer_head is pointing to the corresponding file inode structure. In the function filp_close(); filp->f_path.dentry->d_inode is actually giving the inode of the file which is going to close. Also b_page of the struct bio points to the corresponding buffer head. These facts give an opportunity to pass the inode as well as file close information till the generic block layer. In the mmc_queue_thread; struct request contains a pointer to the bio and hence our structure element “cid” can be obtained through (( struct buffer_head *)(req->bio->bi_private))->cid. In addition to the file data, meta data also passed to the drivers and we found that the flag “REQ_META” of blk_types.h can be used to differentiate among normal data requests and metadata requests.

So now different contexts can be opened with the inode numbers or REQ_META flag. The context can be closed based on the file_close flag of the structure context_id.

IV. HOW THE PROPOSAL IMPROVES THROUGHPUT

Inside eMMC there is a NAND and cache. In normal cases after each write requests, cache syncing happens between NAND and cache. This creates some overhead. Context ID gives a provision to update the metadata upon closing a context. This results in reducing some overhead of metadata update over each write requests. The mathematical quantitative analysis for the throughput improvement based on the request sizes, erase blocks and metadata update overhead.

The write units of eMMC are called “erase blocks”. Typically for an erase block of 128 KB write in NAND, we need 4B of memory to store the mapping information. If the eMMC size is 8GB, we will be having 65536 erase blocks. Out of these (65536*4) / (1024*128) = 2 erase blocks will be dedicated in the NAND for storing mapping information. Even if we need to update only 4B, it will take the same time to write one complete erase block. This peculiar behavior of eMMC gives us an opportunity to reduce the metadata update overhead significantly. Typically every 128KB write in a NAND page requires 19200us. For a file write without context ID will create a metadata update overhead of n*19200us, where “n” is the number of merged requests for the particular file. In the latter case it would be a maximum of 2*19200us, under which a file size ≈ 8GB is written.

We took a use case which is writing a video file of 800MB and the histogram of request sizes in the driver level is shown in figure 4.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Request sizes (multiple of .5KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
</tr>
<tr>
<td>1500</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 4 : Histogram of request sizes in an 800 MB write case

Total metadata update overhead of without context-ID case can be calculated as sum of overheads of each of the request sizes. So

Overhead without context-ID = n*19200us
= 3777*19200us
= 72.52 Seconds.

Overhead with context-ID = 1*19200us
= .0192Seconds

Time to write 800MB video data = ((800*1024)/128)*.0192s
= 122.88 Seconds

Throughput with Context-ID = 800 / (122.88+.0192)
= 6.51MBPS

Throughput without context-ID = 800/ (122.88+72.52)
= 4.09MBPS

Percentage improvement = (6.51-4.09) / (4.09) *100
= 59.16%

V. CONCLUSION

In this paper we investigated the current storage performance bottleneck in Android phones and proposes an inode based context management for eMMC 4.5 which can improve storage throughput. A quantitative analysis of throughput improvement in a use case of 800MB video write shows an improvement of 59.16%.
The improvement depends upon various factors such as write size, erase block size, number of requests in the driver for a given write case, and request sizes for the same. Presently the proposal is implemented for widespread ext4 filesystem. This proposal can be enhanced by incorporating other filesystems also such as Btrfs.

REFERENCES


+++
Investigation of Techniques for Efficient & Accurate Indexing for Scalable Record Linkage & Deduplication

Sunitha Yeddula & K. Lakshmaiah
Dept of CSE, Madanapalle Institute of Technology and Science, Madanapalle
E-mail : Yeddulasunitha08@gmail.com

Abstract – Record linkage is the process of matching records from several databases that refer to the same entities. When applied on a single database, this process is known as deduplication. Increasingly, matched data are becoming important in many applications areas, because they can contain information that is not available otherwise, or that is too costly to acquire. Removing duplicate records in a single database is a crucial step in the data cleaning process, because duplicates can severely influence the outcomes of any subsequent data processing or data mining. With the increasing size of today’s databases, the complexity of the matching process becomes one of the major challenges for record linkage and deduplication. In recent years, various indexing techniques have been developed for record linkage and deduplication. They are aimed at reducing the number of record pairs to be compared in the matching process by removing obvious non-matching pairs, while at the same time maintaining high matching quality. This paper presents a survey of variations of six indexing techniques. Their complexity is analyzed, and their performance and scalability is evaluated within an experimental framework using both synthetic and real data sets. These experiments highlight that one of the most important factors for efficient and accurate indexing for record linkage and deduplication is the proper definition of blocking keys.

Keywords - Data matching, data linkage, entity resolution, index techniques, blocking, experimental evaluation, scalability.

I. INTRODUCTION

As many businesses, government agencies and research projects collect increasingly large amounts of data, techniques that allow efficient processing, analyzing and mining of such massive databases have in recent years attracted interest from both academia and industry. One task that has been recognized to be of increasing importance in many application domains is the matching of records that relate to the same entities from several databases. Often, information from multiple sources needs to be integrated and combined in order to improve data quality, or to enrich data to facilitate more detailed data analysis. The records to be matched frequently correspond to entities that refer to people, such as clients or customers, patients, employees, tax payers, students, or travelers.

The task of record linkage is now commonly used for improving data quality and integrity, to allow re-use of existing data sources for new studies, and to reduce costs and efforts in data acquisition. In the health sector, for example, matched data can contain information that is required to improve health policies, information that traditionally has been collected with time consuming and expensive survey methods[5],[6]. Linked data can also help in health surveillance systems to enrich data that is used for the detection of suspicious patterns.

Statistical agencies have employed record linkage for several decades on a routinely basis to link census data for further analysis. Many businesses use deduplication and record linkage techniques with the aim to deduplicate their databases to improve data quality or compile mailing lists, or to match their data across organizations. Many government organizations are now increasingly employing record linkage, for example within and between taxation offices and departments of social security to identify people who register for assistance multiple times, or who work and collect unemployment benefits, at the same time who are unemployed and handicapped. Other domains where record linkage is of high interest are fraud and crime detection, as well as national security. Security agencies and crime investigators increasingly rely on the ability to quickly access files for a particular individual under investigation, or crosscheck records from disparate databases, which may help to prevent crimes and terror by early intervention.

The problem of finding records that relate to the same entities not only applies to databases that contain information about people. Other types of entities that
sometimes need to be matched include records about businesses, consumer products, publications and bibliographic citations, Web pages, Web search results, or genome sequences. In bioinformatics, for example, record linkage techniques can help find genome sequences in large data collections that are similar to a new, unknown sequence. In the field or information retrieval, it is important to remove duplicate documents (such as Web pages and bibliographic citations) in the results returned by search engines, in digital libraries or in automatic text indexing systems [7],[8]. Another application of growing interest is finding and comparing consumer products from different online stores.

In situations where unique entity identifiers (or keys) are available across all the databases to be linked, the problem of matching records at the entity level becomes trivial: a simple database join is all that is required. However, in most cases no such unique identifiers are shared by all databases, and more sophisticated linkage techniques are required. While statisticians and health researchers commonly name the task of matching records as data or record linkage, the computer science and database communities refer to the same process as data or field matching, data integration, data scrubbing or cleaning, data cleansing, duplicate detection, information integration, entity resolution, reference reconciliation, or as the merge/purge problem.

In commercial processing of business mailing lists and customer databases, record linkage is usually seen as a component of ETL (extraction, transformation and loading) tools.

A. The Record Linkage Process

![Outline of the general record linkage process.](image)

The indexing step generates candidate record pairs, while the outputs of the comparison step are vectors containing numerical similarity values.

Figure 1 outlines the general steps involved in the linking of two databases. Because most real-world data are dirty and contain noisy, incomplete and incorrectly formatted information,

First step: in any record linkage or deduplication project is data cleaning and standardization [1].

The main task of data cleaning and standardization is the conversion of the raw input data into well defined, consistent forms, as well as the resolution of inconsistencies in the way information is represented and encoded.

Second step: (‘Indexing’) is the topic of this survey, in which the indexing step generates pairs of candidate records.

These records are compared in detail in the comparison step using a variety of comparison functions appropriate to the content of the record fields (attributes). Approximate string comparisons, which take (typographical) variations into account, are commonly used on fields that for example contain name and address details, while comparison functions specific for date, age, and numerical values are used for fields that contain such data. Several fields are normally compared for each record pair, resulting in a vector that contains the numerical similarity values calculated for that pair. The next step in the record linkage process is to classify the compared candidate record pairs into matches, non-matches, and possible matches, depending upon the decision model used. Record pairs that were removed in the indexing step are classified as non-matches without being compared explicitly. If record pairs are classified into possible matches, a clerical review process is required where these pairs are manually assessed and classified into matches or no matches. Measuring and evaluating the quality and complexity of a record linkage project is a final step in the record linkage process.

II. INDEXING FOR RECORD LINKAGE AND DEDUPLICATION

When two databases, A and B, are to be matched, potentially each record from A needs to be compared with every record from B, resulting in a maximum number of \(|A| \times |B|\) comparisons between two records. Similarly, when deduplicating a single database A, the maximum number of possible comparisons is \(|A| \times (|A| - 1)/2\), because each record in A potentially needs to be compared with all other records. The performance bottleneck in a record linkage or deduplication system is usually the expensive detailed comparison of field (attribute) values between records, making the naive
approach of comparing all pairs of records not feasible when the databases are large.

At the same time, assuming there are no duplicate records in the databases to be matched, then the maximum possible number of true matches will correspond to \(\min(|A|, |B|)\). Similarly, for a deduplication the number of unique entities (and thus true matches) in a database is always smaller than or equal to the number of records in it. Therefore, while the computational efforts of comparing records increase quadratically as databases are getting larger, the number of potential true matches only increases linearly in the size of the databases.

It is clear that the vast majority of comparisons will be between records that are not matches. The aim of the indexing step is to reduce this large number of potential comparisons by removing as many record pairs as possible that correspond to non matches. The traditional record linkage approach\[3],[4\] has employed an indexing technique commonly called blocking\[2\], which splits the databases into non-overlapping blocks, such that only records within each block are compared with each other. A blocking criterion, commonly called a blocking key, is either based on a single record field (attribute), or the concatenation of values from several fields. An important criteria for a good blocking key is that it can group similar values into the same block. What constitutes a ‘similar’ value depends upon the characteristics of the data to be matched. Similarity can refer to similar sounding or similar looking values based on phonetic or character shape characteristics.

Several important issues need to be considered when record fields are selected to be used as blocking keys. The first issue is that the quality of the values in these fields will influence the quality of the generated candidate record pairs. Ideally, fields containing the fewest errors, variations or missing values should be chosen. Any error in a field value used to generate a BKV will potentially result in records being inserted into the wrong block, thus leading to missing true matches.

A second issue that needs to be considered when defining blocking keys is that the frequency distribution of the values in the fields used for blocking keys will affect the size of the generated blocks. Often this will be the case even after phonetic or other encodings have been applied. The largest blocks generated in the indexing step will dominate execution time of the comparison step, because they will contribute a large portion of the total number of candidate record pairs. Therefore, it is of advantage to use fields that contain uniformly distributed values because they will result in blocks of equal sizes.

When blocking keys are defined, there is also a tradeoff that needs to be considered. On one hand, having a large number of smaller blocks will result in fewer candidate record pairs that will be generated. This will likely increase the number of true matches that are missed. On the other hand, blocking keys that result in larger blocks will generate an increased number of candidate record pairs that likely will cover more true matches, at the cost of having to compare more candidate pairs.

### III. INDEXING TECHNIQUES

In this section, the traditional blocking approach and five more recently developed indexing techniques and variations of them are discussed in more detail.

The estimated number of candidate record pairs will be calculated for two different frequency distributions of BKVs. The first assumes a uniform distribution of values, resulting in each block containing the same number of records. The second assumes that the frequencies of the BKVs follow Zipf’s law\[9\], a frequency distribution that is commonly found in data sets that contain values such as personal names. Zipf’s law states that in a list of words ranked according to their frequencies, the word at rank \(r\) has a relative frequency that corresponds to \(1/r\). Conceptually, the indexing step of the record linkage process can be split into the following two phases:

1) **Build** - All records in the database (or databases) are read, their BKVs are generated, and records are inserted into appropriate index data structures. For most indexing techniques, an inverted index can be used. The BKVs will become the keys of the inverted index, and the record identifiers of all records that have the same BKV will be inserted into the same inverted index list. Figure 2 illustrates this for a small example data set.

<table>
<thead>
<tr>
<th>Identifiers</th>
<th>Surnames</th>
<th>BKVs (Soundex encoding)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Smith</td>
<td>S530</td>
</tr>
<tr>
<td>R2</td>
<td>Miller</td>
<td>M460</td>
</tr>
<tr>
<td>R3</td>
<td>Peters</td>
<td>P362</td>
</tr>
<tr>
<td>R4</td>
<td>Myler</td>
<td>M460</td>
</tr>
<tr>
<td>R5</td>
<td>Smyth</td>
<td>S530</td>
</tr>
<tr>
<td>R6</td>
<td>Millar</td>
<td>M460</td>
</tr>
<tr>
<td>R7</td>
<td>Smyth</td>
<td>S530</td>
</tr>
<tr>
<td>R8</td>
<td>Miller</td>
<td>M460</td>
</tr>
</tbody>
</table>
When linking two databases, either a separate index data structure is built for each database, or a single data structure with common key values is generated. For the second case, each record identifier needs to include a flag that indicates from which database the record originates.

2) Retrieve - For each block, its list of record identifiers is retrieved from the inverted index, and candidate record pairs are generated from this list. For a record linkage, all records in a block from one database will be paired with all records from the block with the same BKV from the other database, while for a deduplication each record in a block will be paired with all other records in the same block. For example, from the block with key ‘S530’ from Figure 2 the pairs (R1, R5), (R1, R7) and (R5, R7) will be generated.

A. Traditional Blocking

This technique has been used in record linkage since the 1960s [4]. All records that have the same BKV are inserted into the same block, and only records within the same block are then compared with each other. Each record is inserted into one block only (assuming a single blocking key definition). Traditional blocking can be implemented efficiently using a standard inverted index [9], the identifiers of all records in the same block are retrieved and the corresponding candidate record pairs are generated. While traditional blocking does not have any explicit parameters, the way blocking keys are defined will influence the quality and number of candidate record pairs that are generated.

A major drawback of traditional blocking is that errors and variations in the record fields used to generate BKVs will lead to records being inserted into the wrong block. This drawback can be overcome by using several blocking key definitions based on different record fields, or different encodings applied on the same record fields. A second drawback of traditional blocking is that the sizes of the blocks generated depend upon the frequency distribution of the BKVs, and thus it is difficult in practice to predict the total number of candidate record pairs that will be generated.

B. Sorted Neighbourhood Indexing

This technique was first proposed in the mid 1990s [10]. Its basic idea is to sort the database(s) according to the BKVs, and to sequentially move a window of a fixed number of records w (w > 1) over the sorted values. Candidate record pairs are then generated only from records within a current window.

1) Sorted Array Based Approach

In this first approach, as originally proposed, the BKVs are inserted into an array that is sorted alphabetically. The window is then moved over this sorted array and candidate record pairs are generated from all records in the current window. In case of a record linkage, the BKVs from both databases will be inserted into one combined array and then sorted alphabetically, but candidate record pairs are generated in such a way that for each pair one record is selected from each of the two databases.

2) Inverted Index Based Approach

It is an alternative approach [11] for the sorted neighborhood. Rather than inserting BKVs into a sorted array, this approach utilizes an inverted index similar to traditional blocking. The index keys contain the alphabetically sorted BKVs, the window is moved over these sorted BKVs, and candidate record pairs are formed from all records in the corresponding index lists. Similar to the sorted array based approach, most candidate record pairs are generated in several windows, but each unique candidate pair will again only be compared once in the comparison step. The number of generated candidate record pairs with this approach depends upon the number of record identifiers that are stored in the inverted index lists.

3) Adaptive Sorted Neighbourhood Approach

Recent research has looked at how the sorted neighbourhood indexing technique based on a sorted array can be improved. The issue of having a fixed block size w which can result in missed true matches (because not all same BKVs fit into one window) has been addressed through an adaptive approach to dynamically set the window size. Due to the adaptive nature of the approach, where block sizes are determined by the similarities between BKVs.

C. Q-gram Based Indexing

This technique aims to allow for ‘fuzzy’ blocking, by converting the blocking key values into lists of q-grams (sub-strings of length q), and, based on sub-lists
of these q-gram lists, each record is inserted into several blocks according to a Jaccard - based similarity threshold. While this technique improves entity resolution for data that contains a large proportion of errors and modifications, its computational complexity makes it unsuitable for large databases.

D. Suffix Array Based Indexing

The basic idea of this suffix array based indexing technique [12] is to insert the blocking key values and their suffixes into a suffix array based inverted index. A suffix array contains strings or sequences and their suffixes in an alphabetically sorted order. Similar to canopy clustering, each record might be inserted into several blocks, depending upon the length of their blocking key values. Record pairs will then be formed from all pairs that are in the same inverted index list.

1) Robust Suffix Array Based Indexing

An improvement upon the original suffix array based indexing technique has recently been proposed. Similar to adaptive blocking, the inverted index lists of suffix values that are similar to each other in the sorted suffix array are merged. An approximate string similarity measure is calculated for all pairs of neighboring suffix values, and if the similarity of a pair is above a selected threshold t, then their lists are merged to form a new larger block.

E. Canopy Clustering

This indexing technique is based on the idea of using a computationally cheap clustering approach to create high-dimensional overlapping clusters, from which blocks of candidate record pairs can then be generated[13],[14]. Clusters are created by calculating the similarities between BKVs using measures such as Jaccard or TF-IDF/cosine. Both of these measures are based on tokens, which can be characters, qgrams or words. They can be implemented efficiently using an inverted index which has tokens, rather than the actual BKVs, as index keys.

1) Threshold Based Approach

In this originally proposed approach [13],[14], two similarity thresholds are used to create the overlapping clusters. All records rx that are within a loose similarity, tl, to rc are inserted into the current cluster (e.g. all records with tl ≤ sJ ). Of these, all records that are within a tight similarity threshold tt (with tt ≥ tl), will be removed from the pool of candidate records. This process of randomly selecting a centroid record rc, calculating the similarities between this and all other records in the pool, and inserting records into clusters, is repeated until no candidate records are left in the pool. If tl = tt, the clusters will not be overlapping, which means each record will be inserted into one cluster only. If both tl = 1 and tt = 1 (i.e. exact similarity only), canopy clustering will generate the same candidate record pairs as traditional blocking.

2) Nearest Neighbour Based Approach

An alternative to using two thresholds is to employ a nearest neighbor based approach to create the overlapping clusters. The idea is to replace the two threshold parameters, tl and tt, with two nearest neighbour parameters, nl and nt (with nl ≥ nt). The first parameter, nl, corresponds to the number of record identifiers that are inserted into each cluster, while nt is the number of record identifiers that are removed from the pool of candidate records in each step of the algorithm. Similar to the threshold based approach, the process of creating overlapping clusters starts by randomly selecting a record rc from the pool of initially all records. Similarities are then calculated between the rc and the records rx that have tokens in common in the inverted index. The nl records closest to rc are inserted into the current cluster, and of these the nt records closest to rc are removed from the pool.

This approach will result in all clusters containing nl record identifiers, independently of the frequency distribution of the BKVs. Therefore, blocks of uniform size will be created, allowing the calculation of the number of generated record pairs. The number of clusters only depends upon the number of records in the database(s) to be matched or deduplicated, and the values of nl and nt. The number of clusters generated corresponds to nA/nt and nB/nt, respectively, and each cluster will contain nl records.

F. String-Map Based Indexing

This indexing technique [15] is based on mapping BKVs (assumed to be strings) to objects in a multi-dimensional euclidean space, such that the distances between pairs of strings are preserved. Any string similarity measure that is a distance function can be used in the mapping process. Groups of similar strings are then generated by extracting objects in this space that are similar to each other. The approach is based on a modification of the FastMap algorithm, called StringMap that has a linear complexity in the number of strings to be mapped. Similar to canopy clustering based indexing; overlapping clusters can be extracted from the multidimensional grid index. An object (referring to a BKV) is randomly picked from the pool of (initially all) objects in the grid based index, and the objects in the same, as well as in neighbouring grid cells, are retrieved from the index. Similar to canopy clustering, either two thresholds, tl and tt, or the number of nearest neighbours, nl and nt, can be used to insert similar objects into clusters, and remove objects from the pool.
with a similarity larger than \( t_t \), or that are the \( n_t \) nearest objects to the centroid object.

**IV. EXPERIMENTAL EVALUATIONS**

The aim of the experiments conducted was to evaluate the presented indexing techniques within a common framework, to answer questions such as: How do parameter values and the choice of the blocking key influence the number and quality of the candidate record pairs generated? How do indexing techniques perform with different types of data? Which indexing techniques show better scalability to larger databases?

**A. Test Data Sets**

Two series of experiments were conducted, the first using four ‘real’ data sets that have previously been used by the record linkage research community, and the second using artificial data sets. Table 3 summarizes these data sets. The aim of the first series of experiments was to investigate how different indexing techniques are able to handle various types of data, while the second series was aimed at investigating the scalability of the different indexing techniques to larger data sets. The first three ‘real’ data sets were taken from the Second String toolkit1. ‘Census’ contains records that were generated by the US Census Bureau based on real census data; ‘Cora’ contains bibliographic records of machine learning publications; and ‘Restaurant’ contains records extracted from the Fodor and Zagat restaurant guides. The ‘CDDB’ data set contains records of audio CDs, such as their title, artist, genre and year. This last data set was recently used in the evaluation of a novel indexing technique. The true match status of all record pairs is available in all four data sets.

Artificial data sets were generated using the Febrl data generator. This generator first creates original records based on frequency tables that contain real name and address values, as well as other personal attributes; followed by the generation of duplicates of these records based on random modifications such as inserting, deleting or substituting characters, and swapping, removing, inserting, splitting or merging words.

Table 1

<table>
<thead>
<tr>
<th>Data set name</th>
<th>Task</th>
<th>Number of records</th>
<th>Total number of true matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Census</td>
<td>Linkage</td>
<td>449+392</td>
<td>327</td>
</tr>
<tr>
<td>Restaurant</td>
<td>Deduplication</td>
<td>864</td>
<td>112</td>
</tr>
</tbody>
</table>

As shown in Table, two series of artificial data sets were created. The ‘Clean’ data contain 80% original and 20% duplicate records, with up to three duplicates for one original record, a maximum of one modification per attribute, and a maximum of three modifications per record. The ‘Dirty’ data contain 60% original and 40% duplicate records, with up to nine duplicates per original record, a maximum of three modifications per attribute, and a maximum of ten modifications per record.

**B. Quality and Complexity Measures**

Four measures are used to assess the complexity of the indexing step and the quality of the resulting candidate record pairs [9], [10]. The total number of matched and non-matched record pairs are denoted with \( n_M \) and \( n_N \), respectively, with \( n_M + n_M = n_A 	imes n_B \) for the linkage of two databases, and \( n_M + n_N = n_A(n_A - 1)/2 \) for the deduplication of one database. The number of true matched and true non-matched candidate record pairs generated by an indexing technique is denoted with \( s_M \) and \( s_N \), respectively, with \( s_M + s_N \leq n_M + n_N \).

The reduction ratio, \( RR = 1.0 - (s_M + s_N) / (n_M + n_N) \), measures the reduction of the comparison space, i.e. the fraction of record pairs that are removed by an indexing technique. The higher the RR value, the less candidate record pairs are being generated.

Table 2

<table>
<thead>
<tr>
<th>Indexing techniques</th>
<th>Label used in figures</th>
<th>Time in milli-seconds per candidate record pair</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Minimum</td>
</tr>
<tr>
<td>Traditional blocking</td>
<td>TB1o</td>
<td>1</td>
</tr>
<tr>
<td>Sorted neighborhood</td>
<td>Array based</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Inverted Index</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Adaptive</td>
<td>8</td>
</tr>
<tr>
<td>Q-Gram</td>
<td>Q-gram based indexing</td>
<td>4</td>
</tr>
</tbody>
</table>

The label used in the result figures, the number of different parameter setting evaluated, and the run-times in milli-seconds per candidate pair required to build each of the evaluated indexing techniques.
Canopy clustering

<table>
<thead>
<tr>
<th></th>
<th>CaTh</th>
<th>CaNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold</td>
<td>8</td>
<td>0.003</td>
</tr>
<tr>
<td>Nearest</td>
<td>8</td>
<td>0.004</td>
</tr>
<tr>
<td>Neighborhood</td>
<td>1.151</td>
<td>1.912</td>
</tr>
</tbody>
</table>

String map

<table>
<thead>
<tr>
<th></th>
<th>STMTh</th>
<th>StMNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold</td>
<td>32</td>
<td>0.004</td>
</tr>
<tr>
<td>Nearest</td>
<td>32</td>
<td>0.018</td>
</tr>
<tr>
<td>Neighborhood</td>
<td>21.715</td>
<td>19.386</td>
</tr>
</tbody>
</table>

Suffix Array

<table>
<thead>
<tr>
<th></th>
<th>SuAr</th>
<th>SuArSu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suffix Array</td>
<td>6</td>
<td>0.024</td>
</tr>
<tr>
<td>Nearest</td>
<td>1.119</td>
<td>3.542</td>
</tr>
<tr>
<td>Neighborhood</td>
<td>11.498</td>
<td>28.082</td>
</tr>
</tbody>
</table>

Robust RoSuA 48 0.010 0.434 0.856 10.421

However, reduction ratio does not take the quality of the generated candidate record pairs into account (how many are true matches or not). Pairs completeness, PC = sM+nM, is the number of true matched candidate record pairs generated by an indexing technique divided by the total number of true matched pairs. Finally, pairs quality, PQ = sM/sM+sN, is the number of true matched candidate record pairs generated by an indexing technique divided by the total number of candidate pairs generated. A high PQ value means an indexing technique is efficient and generates mostly true matched candidate pairs. On the other hand, a low PQ value means a large number of non-matches are also generated.

V. CONCLUSION

The number of candidate record pairs generated by these techniques has been estimated and their efficiency and scalability has been evaluated using various data sets. These experiments highlight that one of the most important factors for efficient and accurate indexing for record linkage and deduplication is the proper definition of blocking keys. Because training data in the form of known true matches and non-matches is often not available in real world applications. The indexing techniques in this investigation are heuristic approaches that aim to split the records in a database into blocks such that matches are inserted in to the same block and non-matches in to different blocks.

ACKNOWLEDGMENT

Students work is incomplete until they thank the almighty & his teachers. I sincerely believe in this and would like to thank Mr. K. Lakshmaiah, M. Tech(Ph.D), Associate prof., of the department of Computer Science & Engineering, MITS Madanapalle, for his encouragement, motivation and for guiding me to write this paper. Also I am grateful to Dr. A. Nagaraja Rao, M. Tech., Ph.D. Prof., in CSE and Head Of the Department of Computer Science, MITS, Madanapalle.

REFERENCES


Text Clustering For Web Document Using
Concept–Based Mining Model

N. Divya Jyothi¹, S.Kusuma² & P.Bindu Madhavi³
¹&³Dept of CSE, MITS, Madanapalle
³Dept of CSE, ACEM, Madanapalle
E-mail : divya1214jyothi@yahoo.com¹, Kusu87@gmail.com², bindubonthala@gmail.com³

Abstract – A new concept-based similarity measure which makes use of the concept analysis on the sentence, document, corpus levels and on web documents is proposed. This similarity measure out performs other similarity measures that are based on term analysis models of the document only. The similarity between documents is based on a combination of sentence-based, document-based, corpus-based and web document–based concept analysis. Similarity based on matching of concepts between document pairs, is shown to have a more significant effect on the clustering quality due to the similarity’s insensitivity to noisy terms that can lead to an incorrect similarity. The concepts are less sensitive to noise when it comes to calculating document similarity. Usually, in text mining techniques, the term frequency of a term (word or phrase) is computed to explore the importance of the term in the web document. However, two terms can have the same frequency in their documents, but one term contributes more to the meaning of its sentences than the other term.

Keywords - Concept-based mining model, Sentence-based, Document-based, Corpus based, Concept analysis, Conceptual term frequency, Concept-based similarity.

I. INTRODUCTION

The process of grouping a set of physical objects into classes of similar objects is called clustering. Cluster is a collection of data objects that are similar to one another with in the same cluster and are dissimilar to the objects in other classes. Cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression. Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their similarity.

Information Retrieval (Information Access)

– No genuinely new information is found.
– The desired information merely coexists with other valid pieces of information.

Computation Linguistics (CPL) & Natural Language Processing (NLP)

– An extrapolation from Data Mining on numerical data to Data Mining from textual collections [Hearst 1999].

Step1: TEXT

Document Clustering:-

• Large volume of textual data
• Most popular Document Clustering methods are:
  – K-Means clustering.
  – Agglomerative hierarchical clustering.
• Several input modes
Text Clustering For Web Document Using Concept-Based Mining Model

Text is intended for different consumers, i.e. different languages (human consumers) and different formats (automated consumers).

- **Dependency**
  - Words and phrases create context for each other.

- **Ambiguity**
  - Word ambiguity.
  - Sentence ambiguity.

- **Noisy data**
  - Erroneous data.
  - Misleading (intentionally) data.

- **Unstructured text**
  - Chat room, normal speech ...

- **High dimensionality (sparse input)**
  - Tens of thousands of words (attributes).
  - Only a very small percentage is used in a typical document.
  - For example:
    - Top 2 words »10-15% all word occurrences.
    - Top 6 words »20% of all word occurrences.
    - Top 50 words »50% of all occurrences

**Step 2: Text Preprocessing**

- **Text cleanup**
  - e.g., remove ads from web pages, normalize text converted from binary formats, deal with tables, figures and formulas …

- **Tokenization**
  - Splitting up a string of characters into a set of tokens.
  - Need to deal with issues like:
    - Apostrophes, e.g., “John’s sick”, is it 1 or 2 tokens?
    - Hyphens, e.g., database vs. data-base vs. data base.
    - How should we deal with “C++”, “A/C”, “:-)”, “…”?
    - Is the amount of white spaces significant?

- **Parts Of Speech tagging**
  - The process of marking up the words in a text with their corresponding parts of speech.
  - Rule based
  - Depends on grammatical rules.
  - Statistically based
  - Relies on different word order probabilities.
  - Needs a manually tagged corpus for machine learning.

- **Word Sense Disambiguation**
  - Determining in which sense a word having a number of distinct senses is used in a given sentence.
  - “The king saw the rabbit with his glasses”

**Semantic Structures:**

- Two methods:
  - Full parsing: Produces a parse tree for a sentence.

**Step 3: Text Transformation and Attribute Generation**

- **Text Representation:**
  - Text document is represented by the words (features) it contains and their occurrences.

- **Feature Selection:**
  - Select just a subset of the features to represent a document.
  - Can be viewed as creating an improved text representation.

- **Why do it?**
  - Many features have little information content e.g. stop words.
  - Some features are misleading
  - Some features are redundant
  - Independence assumptions result in double-counting
  - Some algorithms work better with small feature sets e.g. because they create complex classifiers…

So the space of possible classifiers is very large

- **Stop words removal**
  - The most common words are unlikely to help text mining, e.g., “the”, “a”, “an”, “you”…
• Stemming
  – Identifies a word by its root.
  – Reduce dimensionality (number of features).
  – e.g. flying, flew → fly
  – Two common algorithms:
    • Porter’s Algorithm.
    • KSTEM Algorithm.
• Stemming Examples
  – Original Text
    Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals.
    – Porter Stemmer (stop words removed)
    – KSTEM (stop words removed)

Step 4: Attribute selection
• Further reduction of dimensionality
  – Learners have difficulty addressing tasks with high dimensionality.
  – Scarcity of resources and feasibility issues also call for a further cutback of attributes.
• Irrelevant features
  – Not all features help!
  – e.g., the existence of a noun in a news article is unlikely to help classify it as “politics” or “sport”.

Step 5: Data Mining
• At this point the Text mining process merges with the traditional Data Mining process.
• Classic Data Mining techniques are used on the structured database that resulted from the previous stages.

Step 6: Interpretation and Evaluation
• This is a purely application-dependent stage.
  – Terminate
  – Results well-suited for application at hand.
  – Iterate
  – Results not satisfactory but significant.
  – The results generated are used as part of the input for one or more earlier stages.

II. RELATED WORK

NATURAL Language Processing (NLP) is both a modern computational technology and a method of investigating and evaluating claims about human language itself. NLP is a term that links back into the history of Artificial Intelligence (AI), the general study of cognitive function by computational processes, with an emphasis on the role of knowledge representations. Text mining attempts to discover new, previously unknown information by applying techniques from natural language processing and data mining.

Clustering, one of the traditional data mining techniques, is an unsupervised learning paradigm where clustering methods try to identify inherent groupings of the text documents, so that a set of clusters is produced in which clusters exhibit high intracluster similarity and low inter cluster.

In text mining techniques, the term frequency of a term (word or phrase) is computed to explore the importance of the term in the document. However, two terms can have the same frequency in their documents, but one term contributes more to the meaning of its sentences than the other term. It is important to note that extracting the relations between verbs and their arguments in the same sentence has the potential for analyzing terms within a sentence. The proposed model captures the semantic structure of each term within a sentence and document rather than the frequency of the term within a document only. In this model, three measures for analyzing concepts on the sentence, document, and corpus levels are computed. Each sentence is labeled by a semantic role labeler that determines the terms which contribute to the sentence semantics associated with their semantic roles in a sentence. Each term that has a semantic role in the sentence, is called a concept. Concepts can be either words or phrases and are totally dependent on the semantic structure of the sentence. When a new document is introduced to the system, the proposed mining model can detect a concept match from this document to all the previously processed documents in the data set by scanning the new document and...
extracting the matching concepts. A new concept-based similarity measure which makes use of the concept analysis on the sentence, document, and

Corpus levels are proposed. This similarity measure outperforms other similarity measures that are based on term analysis models of the document only. The similarity between documents is based on a combination of sentence-based, document-based, and corpus-based concept analysis. Similarity based on matching of concepts between document pairs, is shown to have a more significant effect on the clustering quality due to the similarity’s insensitivity to noisy terms that can lead to an incorrect similarity. The

Concepts are less sensitive to noise when it comes to calculating document similarity. This is due to the fact that these concepts are originally extracted by the semantic role Labeler and analyzed with respect to the sentence, document, and corpus levels. Thus, the matching among these concepts is less likely to be found in nonrelated documents.

Verb argument structure: (e.g., John hits the ball), “hits” is the verb. “John” and “the ball” are the arguments of the verb “hits,”

III. PROPOSED WORK
Concept-based Mining Model:

The proposed mining model is an extension of the work in [22]. The proposed concept-based mining model consists of sentence-based concept analysis, document-based concept analysis, corpus-based concept-analysis, and concept-based similarity measure, as depicted in Fig. 1

![Fig.1.concept based mining model system](image)

A raw text document is the input to the proposed model. Each document has well-defined sentence boundaries. Each sentence in the document is labeled automatically. The labeled verb argument structures, the output of the role labeling task, are captured and analyzed by the concept-based mining model on sentence, document and corpus levels. In this model, both the verb and the argument are considered as terms. One term can be an argument to more than one verb in the same sentence. This means that this term can have more than one semantic role in the same sentence. In such cases, this term plays important semantic roles that contribute to the meaning of the sentence. In the concept-based mining model, a labeled terms either word or phrase is considered as concept. The objective behind the concept-based analysis task is to achieve an accurate analysis of concepts on the sentence, document, and corpus levels rather than a single-term analysis on the document only.

Sentence-Based Concept Analysis

To analyze each concept at the sentence level, a new concept-based frequency measure, called the conceptual term frequency $\text{ctf}$ is proposed. The $\text{ctf}$ calculations of concept $c$ in sentence $s$ and document $d$ are as follows:

3.1.1 Calculating $\text{ctf}$ of Concept $c$ in Sentence $s$

The $\text{ctf}$ is the number of occurrences of concept $c$ in verb argument structures of sentence $s$. The concept $c$, which frequently appears in different verb argument structures of the same sentence $s$, has the principal role of contributing to the meaning of $s$. In this case, the $\text{ctf}$ is a local measure on the sentence level.

3.1.2 Calculating $\text{ctf}$ of Concept $c$ in Document $d$

A concept $c$ can have many $\text{ctf}$ values in different sentences in the same document $d$. Thus, the $\text{ctf}$ value of concept $c$ in document $d$ is calculated by:

$$\text{ctf} = \sum_{s} \text{ctf}_{sn}$$

Where $sn$ is the total number of sentences that contain concept $c$ in document $d$.

Document-Based Concept Analysis

To analyze each concept at the document level, the concept based term frequency $\text{tf}$, the number of occurrences of a concept (word or phrase) $c$ in the original document, is calculated. The $\text{tf}$ is a local measure on the document level.

Corpus-Based Concept Analysis

To extract concepts that can discriminate between documents, the concept-based document frequency $\text{df}$, the number of documents containing concept $c$, is calculated. The $\text{df}$ is a global measure on the corpus level.

A Concept-Based Similarity Measure

Concepts convey local context information, which is essential in determining an accurate similarity between documents. A concept-based similarity
measure, based on matching concepts at the sentence, document, corpus and combined approach rather than on individual terms (words) only, is devised. The concept-based measure exploits the information extracted from the concept-based analysis algorithm to better judge the similarity between the documents.

The conceptual term frequency (ctf) is an important factor in calculating the concept-based similarity measure between documents. The more frequent the concept appears in the verb argument structures of a sentence in a document, the more conceptually similar the documents. The concept-based matching consists of either an exact match or partial match between two concepts. Exact match means that both concepts have the same words. Partial match means that one concept includes all the words that appear in the other concept.

**IV. EXPERIMENTAL RESULTS**

To test the effectiveness of concept matching in determining an accurate measure of the similarity between documents, extensive sets of experiments using the concept-based term analysis and similarity measure are conducted.

The similarities which are calculated by using the sentence-based, document-based, corpus-based and the combined approach concept analysis are used to compute four similarity matrices among documents.

Three standard clustering techniques are used for testing the effect of the concept-based similarity on clustering.

**V. CONCLUSION**

In this paper, we have described that the concept-based mining model for enhancing the text clustering for web documents. This procedure is implemented for the static pages and in the proposed model the same concept is applied for the web documents. The web page is converted into the HTML page and the parsing is performed for this web page. By using the concept-based mining model, clustering techniques the similar types of clusters are grouped together. In future work we will search for more robust solution for this problem.

**REFERENCES**


Algorithms for Constructing Edge Magic Total Labeling of Complete Bipartite graphs

Krishnappa H K¹, N K Srinath², S Manjunath³ & Ramakanth Kumar P⁴
¹Dept. of CSE, RVCE., Bangalore, India, ²Dept. of Mathematics, BNMIT, Bangalore, India
³Dept. of ISE, RVCE., Bangalore, India
E-mail : hk_krit@yahoo.co.in¹, srinath nk@yahoo.com², drmanjus@gmail.com³, pramakanth2000@gmail.com⁴

Abstract – The study of graph labeling has focused on finding classes of graphs which admits a particular type of labeling. In this paper we consider a particular class of graphs which demonstrates Edge Magic Total Labeling. The class we considered here is a complete bipartite graph Km,n.

There are various graph labeling techniques that generalize the idea of a magic square has been proposed earlier. The definition of a magic labeling on a graph with v vertices and e edges is a one to one map taking the vertices and edges onto the integers 1,2,3,……., v+e with the property that the sum of the label on an edge and the labels of its endpoints is constant independent of the choice of edge. We use m x n matrix to construct edge magic total labeling of Km,n.

Keywords - Magic squares, Magic constant, Complete graphs, Complete bipartite graph etc.

I. INTRODUCTION

The graph considered is a finite, simple and undirected. Let us consider such a graph G= (V, E). The graph G has a vertex set V=V(G) and edge set E=E(G). We usually denote v=|V| and e=|E|. A general reference for graph theoretic notations is used in this paper.

The labeling of a graph is a map that takes graph elements V or E or V U E to positive numbers. If the domain is only a vertex set then it is called vertex magic. If the domain is only an edge set then the labeling is called edge magic [3]. If the domain consists of both vertices and edges it gives raise to the vertex magic total labeling. The notation of a vertex magic total labeling was introduced in [4,5,6,7].

There are various ways to label the edges of a graph. A connected graph is called a semi magic if there is a labeling of the edges with integers such that for each vertex u the sum of the labels of all edges incident with u is the same for all u. A semi magic labeling where the edges are labeled with distinct positive integers is called magic labeling. A magic labeling is called super magic if the set of edge labels consists of consecutive positive integers [9, 10].

A magic labeling of a graph G(V,E) is called bijection if from V U E to {1,2,…,|V U E|} such that for all edges (x,y) E, f(x)+f(y)+f(x,y)=k, where k is a constant. This is called as edge magic total labeling [EMTL]. They prove that the following graphs are EMTL. Cycle graph Cn is EMTL for all n>2, Complete graph Kn is EMTL iff n=1,2,3,5 or 6 and the complete bipartite graph Km,n is EMTL. The way by which they showed these graphs admits EMTL is trial and error method[12].

In this paper we proved the result [14] that is, all complete bipartite graphs Km,n admits EMTL by using m x n matrix. We proposed an algorithms which takes m and n as inputs and produces a (m+1) x (n+1) matrix which represents the EMTL of Km,n. The magic constant obtained for various complete bipartite graphs are by using EMTL algorithms. Our algorithm finds all the magic constants those lies between minimum and maximum magic constant.

II. CONVENTIONAL REPRESENTATION

The algorithms uses (m+1) x (n+1) matrix to produce EMTL of Km,n. The conventional representation of this matrix for minimum magic constant and for the maximum magic constant is described as below.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>…</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>n+1</td>
<td>mn+m</td>
<td>mn+m</td>
<td>mn+m</td>
<td>mn+m</td>
<td>mn+m</td>
<td></td>
</tr>
<tr>
<td>2(n+1)</td>
<td>……</td>
<td>……</td>
<td>……</td>
<td>……</td>
<td>……</td>
<td>……</td>
</tr>
</tbody>
</table>

International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati 37
Fig. 1 : Matrix representation for magic constant.

Let the matrix is M whose order is \((m+1) \times (n+1)\). The entry for \(M[0,0]=0\), the remaining entries for the 0th row represents the vertex labels for the second partite. Similarly, the remaining entries for the 0th column of \(M\) represents the vertex labels of the first partite. The entries for the remaining cells of the matrix represents the corresponding edge labels of the graph \(K_{m,n}\).

### III. ALGORITHM TO CONSTRUCT MINIMUM MAGIC CONSTANT.

We know that \(K_{m,n}\) has \(m+n+mn\) number of components, hence we need to map numbers from 1,2,...........\(m+n+mn\) to each components of the graph. The algorithm below fills the matrix with these numbers. Finally, we can map these numbers to the corresponding graph elements.

**Algorithm**

**Minimum_Magic_Constant (m, n)**

1. \(M[0,0]\leftarrow 0\)
2. Repeat for \(i\leftarrow 1\) to \(n\) do
   \(M[0,i]\leftarrow i\)
3. Repeat for \(i\leftarrow 1\) to \(m\) do
   \(M[i,0]\leftarrow i*(n+1)\)
4. At this stage the first row and first column were filled with numbers 1,2,...........\(n+1\), 2\((n+1)\), 3\((n+1)\),............\(m(n+1)\).

Let the remaining numbers (\(n+2\), \((n+3)\),........that is the left out numbers from the set \(\{1,2,3,........m+n+mn\}\) are stored in an array \(A\) of size \(m^n\), in descending order.

5. Fill the remaining entries of the matrix \(M\) with the elements of the array \(A\).

\[
k\leftarrow 1
\]

Repeat for \(i\leftarrow 1\) to \(m\) do
Repeat for \(j\leftarrow 1\) to \(n\) do
\(M[i,j]\leftarrow A[k]\)
\(k\leftarrow k+1\)

6. Stop.

Working of this algorithm for \(K_{2,2}\), \(K_{2,3}\) and \(K_{3,2}\) were given below.

**Example 1: \(K_{2,2}\)**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**Example 2: \(K_{2,3}\)**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1**

Magic constant computation.

\(f(u_1)+f(v_1)+f(u_1,v_1)=3+1+8=12\)
\(f(u_1)+f(v_2)+f(u_1,v_2)=3+2+7=12\)
\(f(u_2)+f(v_1)+f(u_2,v_1)=6+1+5=12\)
\(f(u_2)+f(v_2)+f(u_2,v_2)=6+2+4=12\)

Verification by formula: \(mn+m+2n+2\), here \(m=2, n=2\).
\(2*2+2+2*2+2=4+2+4+2=12\).

Mapping of the above table in to the graph is as shown below.

**Example 2: \(K_{3,2}\)**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2**
Magic constant computation.
\[ f(u_1)+f(u_1,v_1)+f(v_1)=4+11+1=16, \]
\[ f(u_1)+f(u_1,v_2)+f(v_2)=4+10+2=16, \]
\[ f(u_1)+f(u_1,v_3)+f(v_3)=4+9+3=16, \]
\[ f(u_2)+f(u_2,v_1)+f(v_1)=8+7+1=16, \]
\[ f(u_2)+f(u_2,v_2)+f(v_2)=8+6+2=16, \]
\[ f(u_2)+f(u_2,v_3)+f(v_3)=8+5+3=16. \]

Verification by formula.
Here \( m=2, n=3 \).
Magic constant = \( mn+m+2n+2 \)
\[ = 2*3+2+2*3+2=6+2+6+2=16. \]

Mapping of the above table in to the graph is as shown below.

\[ \text{Fig. 3} \]

Example 3: \( K_{3,2} \)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>11</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Table. 3

Magic constant computation.
\[ f(u_1)+f(u_1,v_1)+f(v_1)=3+11+1=15, \]
\[ f(u_1)+f(u_1,v_2)+f(v_2)=3+10+2=15, \]
\[ f(u_2)+f(u_2,v_1)+f(v_1)=6+8+1=15, \]
\[ f(u_2)+f(u_2,v_2)+f(v_2)=6+7+2=15, \]
\[ f(u_3)+f(u_3,v_1)+f(v_1)=9+5+1=15, \]
\[ f(u_3)+f(u_3,v_2)+f(v_2)=9+4+2=15. \]

Verification by formula.
Here \( m=3, n=2 \).
Magic constant = \( mn+m+2n+2 \)
\[ = 3*2+3+2*2+2 \]
\[ = 6+3+4+2=15 \]

Mapping of the above table in to the graph is as shown below.

\[ \text{Fig. 4} \]

IV. Algorithm to construct maximum magic constant.
To get the maximum magic constant fill the matrix \( M \) by using the algorithm given below. Finally use the entries of the first row to label the vertices of the second partite and the entries of the first column to label the vertices of the first partite. The remaining entries are used to label the corresponding edges of the graph.

\[ \text{Algorithm} \]

Maximum_Magic_Constant(m, n)
//\( M \) is the global matrix of size at least \( (m+1) \times (n+1) \). Here \( m \) and \( n \) are integers and represents number of vertices in first partite //and second partite of the graph \( Km,n \).

1. [Initialization]
\[ M[0,0] \leftarrow 0 \]
\[ X \leftarrow m+n+m*n \]
2. [Fill the first row with the highest values]
Repeat for \( i \leftarrow 1 \) to \( n \) do
\[ M[0,i] \leftarrow x \]
\[ X \leftarrow x-1 \]
3. [Fill the first column as follows]
\[ M[1,0] \leftarrow x \]
Repeat for \( i \leftarrow 2 \) to \( m \) do
\[ M[i,0] \leftarrow x-(n+1) \]
4. The remaining numbers from the set \{1,2,3,\ldots,m+n+m*n\} which were not used to fill the matrix in step-2, and step-3 store them in an array \(A\) in increasing order.

5. [Fill the remaining entries of the matrix by using the elements of the array \(A\).]

\[
K \leftarrow 1
\]
Repeat for \(i \leftarrow 1\) to \(m\) do
Repeat for \(j \leftarrow 1\) to \(n\) do
\(M[i,j] \leftarrow A[k]\)
\(K \leftarrow k+1\)

6. [Stop]

Working of this algorithm for \(K_{2,2}\), \(K_{2,3}\) and \(K_{3,2}\) were given below.

**Example 1: \(K_{2,2}\)**

Here \(m=2\) and \(n=2\), hence no. of vertices=4 and no. of edges=4.

<table>
<thead>
<tr>
<th>0</th>
<th>8</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4

Magic constant computation.
\(f(u1)+f(v1)+f(u1,v1)=6+8+1=15\)
\(f(u1)+f(v2)+f(u1,v2)=6+7+2=15\)
\(f(u2)+f(v1)+f(u2,v1)=3+8+4=15\)
\(f(u2)+f(v2)+f(u2,v2)=3+7+5=15\)

Verification by formula:
\((n+1)*(2m+1)=(3+1)*(2*2+1)=4*5=20\).

**Mapping of the above table in to the graph is as shown below.**

![Fig. 5](image)

**Example 2: \(K_{2,3}\)**

Here \(m=2\) and \(n=3\), hence no. of vertices=5 and no. of edges=6.

<table>
<thead>
<tr>
<th>0</th>
<th>11</th>
<th>10</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 5

Magic constant computation.
\(f(u1)+f(v1)+f(u1,v1)=8+11+1=20\)
\(f(u1)+f(v2)+f(u1,v2)=8+10+2=20\)
\(f(u1)+f(v3)+f(u1,v3)=8+9+3=20\)
\(f(u2)+f(v1)+f(u2,v1)=4+11+5=20\)
\(f(u2)+f(v2)+f(u2,v2)=4+10+6=20\)
\(f(u2)+f(v3)+f(u2,v3)=4+9+7=20\)

Verification by formula:
\((n+1)*(2m+1)=(3+1)*(2*2+1)=4*5=20\).

**Mapping of the above table in to the graph is as shown below.**

![Fig. 6](image)

IV. CONCLUSION

It has been demonstrated using the two algorithms that the labeling has been successfully done. Construct minimum magic constant algorithm fills the matrix with numbers which could be mapped to the corresponding graph elements. Construction of Maximum magic constant algorithm is used to fill the matrix and the entries that are used to label the corresponding edges of the graph.

References


Data Leakage Detection Using Fake Objects

Sujana Domma & M.SreeDevi
Dept of CSE, Madanapalle Institute of Technology and Science, Madanapalle
E-mail : sujanachowdary.d@gmail.com

Abstract – In a perfect world, there would be no need to hand over sensitive data to agents that may unknowingly or maliciously leak it. And even if we had to hand over sensitive data, in a perfect world, we could watermark each object so that we could trace its origins with absolute certainty. A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data are leaked and found in an unauthorized place. The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Data allocation strategies (across the agents) that improve the probability of identifying leakages. In some cases, we can also inject “realistic but fake” data records to further improve our chances of detecting leakage and identifying the guilty party. The idea is to distribute the data intelligently to agents based on sample data request and explicit data request in order to improve the chance of detecting the guilty agents. we propose the extension of allocation strategies, so that they can handle agents requests in an online fashion.

Keywords - sensitive data, fake objects, data allocation strategies.

I. INTRODUCTION

The rapid growth of internet and related technologies has an unprecedented ability to access and redistribute digital contents. In such a context, enforcing data ownership is an important requirement which requires articulated solutions, encompassing technical, organizational and legal aspects. Though we are still far from such comprehensive solutions, in the last years watermarking techniques have emerged as an important building block which plays a crucial role in addressing the ownership problem. Such techniques allow the owner of the data to embed an imperceptible watermark into the data. A watermark[1][6] describes information that can be used to prove the ownership of data, such as the owner, origin, or recipient of the content.

Traditionally, leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious.

In this paper, we develop a model for assessing the “guilt” of agents. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker. Finally, we also consider the option of adding “fake” objects to the distributed set. That agent was guilty.

II. PROBLEM DEFINITION

Suppose a distributor owns a set \( T = \{t_1, \ldots, t_m\} \) of valuable data objects. The distributor wants to share some of the objects with a set of agents \( U_1, U_2, \ldots, U_n \), but does wish the objects be leaked to other third parties.

An agent receives a subset of objects which belongs to \( T \), determined either by a sample request or an explicit request.

Sample Request = \( \text{SAMPLE}(T, m_i) \) : Any subset records from \( T \) can be given to \( U_i \).

Explicit Request = \( \text{EXPLICIT}(T, \text{condi}) \) : Agent \( U_i \) receives all the \( T \) objects that satisfy condi.

The objects in \( T \) could be of any type and size, e.g., they could be tuples in a relation, or relations in \( R \). After giving objects to agents, the distributor discovers that a set \( S \) of \( T \) has leaked. This means that some third party called the target has been caught in possession of \( S \). For example, this target may be displaying \( S \) on its web site, or perhaps as part of a legal discovery process, the target turned over \( S \) to the distributor.

Since the agents \( U_1, \ldots, U_n \), have some of the data, it is reasonable to suspect them leaking the data. However, the agents can argue that they are innocent, and that the \( S \) data was obtained by the target through other means.
A. Agent Guilt Model

Suppose an agent $U_i$ is guilty if it contributes one or more objects to the target. The event that agent $U_i$ is Guilty for a given leaked set $S$ is denoted by $G_i|S$. $Pr\{G_i|S\}$ is the probability that agent $U_i$ is guilty by given evidence $S$.

To compute $Pr\{G_i|S\}$, estimate the probability that values in $S$ can be “guessed” by the target. For instance, say some of the objects in $t$ are emails of individuals. Conduct an experiment and ask a person may only discover say 20, leading to an estimate of 0.2. Call this estimate as $p$, the probability that object $t$ can be guessed by the target.

The two assumptions[1] regarding the relationship among the various leakage events.

**Assumption 1:** For all $t$, $t \in S$ such that $t \neq S$, the provenance of $t$ is independent of the provenance of $S$.

The term provenance in this assumption statement refers to the source of a value $t$ that appears in the leaked set. The source can be any of the agents who have $t$ in their sets or the target itself.

**Assumption 2:** An object $t \in S$ can only be obtained by the target in one of two ways.

- A single agent leaked $t$ from its own set, or
- The target guessed (or obtained through other means) $t$ without the help of any of the $n$ agents.

To find the probability that an agent $U_i$ is guilty given a set $S$, consider the target guessed $t_1$ with probability $p$ and that agent leaks $t_1$ to $S$ with the probability $1-p$. First compute the probability that he leaks a single object $t$ to $S$. To compute this, define the set of agents $V_i = \{U_i | U_i \in R_i\}$ that have $t$ in their data sets. Then using Assumption 2 and known probability $p$, we have

$$Pr\{\text{some agent leaked } t \text{ to } S\} = 1 - p \quad \ldots \ldots \ldots 1.1$$

Assuming that all agents that belong to $V_i$ can leak $t$ to $S$ with equal probability and using Assumption 2 obtain,

$$Pr\{U_i \text{ leaked } t \to S\} = \begin{cases} 1 - p & \text{if } U_i \in R_i \\ 0 & \text{otherwise} \end{cases} \quad \ldots \ldots \ldots 1.2$$

Given that agent $U_i$ is guilty if he leaks at least one value to $S$, with Assumption 1 and Equation 1.2 compute the probability $Pr\{G_i|S\}$, agent $U_i$ is guilty,

$$Pr\{G_i \mid S\} = \prod_{t \in S \cap R_i} \left(1 - \frac{1-p}{|V_i|}\right) \quad \ldots \ldots \ldots 1.3$$

B. Data Allocation Problem

The distributor “intelligently” gives data to agents in order to improve the chances of detecting a guilty agent. There are four instances of this problem, depending on the type of data requests made by agents and whether “fake objects” [4] are allowed. Agent makes two types of requests, called sample and explicit. Based on the requests the fakes objects are added to data list. Fake objects are objects generated by the distributor that are not in set $T$. The objects are designed to look like real objects, and are distributed to agents together with the $T$ objects, in order to increase the chances of detecting agents that leak data.

![Fig 1: Leakage Problem Instances](image)

The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. Since, fake objects may impact the correctness of what agents do, so they may not always be allowable. Use of fake objects is inspired by the use of “trace” records in mailing lists. The distributor creates and adds fake objects to the data that he distributes to agents. In many cases, the distributor may be limited in how many fake objects he can create.

In $EF$ problems, objective values are initialized by agents’ data requests. Say, for example, that $T = \{t_1, t_2\}$ and there are two agents with explicit data requests such that $R_1 \{ t_1, t_2 \}$ and $R_2 \{ t_1, t_2 \}$. The distributor cannot remove $R_1$ or $R_2$ after the $R_1$ or $R_2$ data to decrease the overlap. However, say the...
distributor can create one fake object \((B = 1)\) and both agents can receive fake object \((b_1 = b_2 = 1)\). If the distributor is able to create more fake objects, he could further improve the objective.

C. Optimization Problem

The distributor’s data allocation to agents has one constraint and one objective. The distributor’s constraint is to satisfy agents’ requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to be able to detect an agent who leaks any portion of his data.

We consider the constraint as strict. The distributor may not deny serving an agent request and may not provide agents with different perturbed versions of the same objects. The fake object distribution as the only possible constraint relaxation.

The objective is to maximize the chances of detecting a guilty agent that leaks all his data objects.

The probability that agent \(U_j\) is guilty if the distributor discovers a leaked table \(S\) that contains all objects is given by the difference functions

\[
\Delta(i, j) \text{ is defined as: }
\]

\[
\Delta(i, j) = \Pr \{G_i \mid R_i\} - \Pr \{G_j \mid R_i\} \quad (1.5)
\]

1) Problem definition: Let the distributor have data requests from \(n\) agents. The distributor wants to give tables \(R_1, \ldots, R_n\) to agents \(U_1, \ldots, U_n\) respectively, so that

- Distribution satisfies agents’ requests; and
- Maximizes the guilt probability differences \(\Delta(i, j)\) for all \(i, j = 1, \ldots, n\) and \(i = j\).

Assuming that the \(R_i\) sets satisfy the agents’ requests, we can express the problem as a multi-criterion

2) Optimization problem:

Maximize \((\ldots, \Delta(i, j), \ldots) \mid R_i \cap R_j \mid \text{ over } R_1, \ldots, R_n\) \quad (1.6)

The approximation [3] of objective of the above equation does not depend on agent’s probabilities and therefore minimize the relative overlap among the agents as

\[
\text{Minimize } (\ldots, \frac{|R_i \cap R_j|}{R_i}) \mid R_i \cap R_j \mid \text{ over } R_1, \ldots, R_n
\]

This approximation is valid if minimizing the relative overlap \(\frac{|R_i \cap R_j|}{R_i}\) maximizes \(\Delta(i, j)\).

D. Objective Approximation

In case of sample request, all requests are of fixed size. Therefore, maximizing the chance of detecting a guilty agent that leaks all his data by minimizing \(|R_i \cap R_j|/R_i\) is equivalent to minimizing \(|R_i \cap R_j|\).

The minimum value of \(|R_i \cap R_j|\) maximizes \(\Pi(R_i \cap R_j)\) and maximizes \(\Delta(i, j)\) since \(|R_i|\) is fixed.

If agents have explicit data requests, then overlaps are defined by their own requests and \(|R_i \cap R_j|\) are fixed. Therefore, minimizing \(|R_i|\) is equivalent to maximizing \(|R_i|\) (with the addition of fake objects). The maximum value of \(|R_i|\) minimizes \(\Pi(R_i)\) and maximizes \(\Delta(i, j)\), since \(\Pi(R_i \cap R_j)\) is fixed.

III. ALLOCATION STRATEGIES

1. In Section A deals with problems with explicit data requests and in Section B with problems with sample data requests.

A. Explicit Data Request

In case of explicit data request with fake not allowed, the distributor is not allowed to add fake objects to the distributed data. So Data allocation is fully defined by the agent’s data request.

In case of explicit data request with fake allowed, the distributor cannot remove or alter the requests \(R\) from the agent. However distributor can add the fake object. In algorithm for data allocation for explicit request, the Input to this is a set of \(R_i, R_2, \ldots, R_n\) from \(n\) agents and different conditions for requests. The e-optimal algorithm finds the agents that are eligible to receiving fake objects. Then create one fake object in iteration and allocate it to the agent selected. The e-optimal algorithm minimizes every term of the objective summation by adding maximum number \(b_i\) of fake objects to every set \(R_i\) yielding optimal solution.

Step 1: Calculate total fake records as sum of fake records allowed.
Step 2: While total fake objects > 0
Step 3: Select agent that will yield the greatest improvement in the sum objective
i.e. i = \arg\max((1|\mathcal{R}_i|)-(1|\mathcal{R}_i|+1))\sigma_{maj}\mathcal{R}_i\cap\mathcal{R}_j
Step 4: Create fake record
Step 5: Add this fake record to the agent and also to fake record set.
Step 6: Decrement fake record from total fake record set.

Algorithm makes a greedy choice by selecting the agent that will yield the greatest improvement in the sum-objective.

B. Sample Data Request:

With sample data requests, each agent \( U_i \) may receive any \( T \) from a subset out of \( |\mathcal{T}| \) different ones. Hence, there are \( \Pi_{i=1} |\mathcal{T}| \) different allocations. In every allocation, the distributor can permute \( T \) objects and keep the same chances of guilty agent detection. The reason is that the guilt probability depends only on which agents have received the leaked objects and not on the identity of the leaked objects. Therefore, from the distributor’s perspective there are different allocations. An object allocation that satisfies requests and ignores the distributor’s objective is to give each agent a unique subset of \( T \) of size \( m \). The s-max algorithm allocates to an agent the data record that yields the minimum increase of the maximum relative overlap among any pair of agents. The s-max algorithm is as follows.

Step 1: Initialize \( \text{Min\_overlap} \leftarrow 1 \), the minimum out of the maximum relative overlaps that the allocations of different objects to \( U_i \).

Step 2: for \( k \in \{k | \mathcal{T}_k \in \mathcal{R}_i \} \) do

Initialize \( \text{max\_rel\_ov} \leftarrow 0 \), the maximum relative overlap between \( \mathcal{T}_k \) to \( U_i \) and any set that the allocation of \( \mathcal{T}_k \) to \( U_i \).

Step 3: for all \( j = 1, \ldots, n : j = i \) and \( \mathcal{T}_k \in \mathcal{R} \) do

Calculate absolute overlap as \( \text{abs\_ov} \leftarrow |\mathcal{R}_i \cap \mathcal{R}_j| + 1 \)

Calculate relative overlap as \( \text{rel\_ov} \leftarrow \text{abs\_ov} / \min(\mathcal{m}_i, \mathcal{m}_j) \)

Step 4: Find maximum relative as \( \text{max\_rel\_ov} \leftarrow \text{MAX} (\text{max\_rel\_ov}, \text{rel\_ov}) \)

If \( \text{max\_rel\_ov} \leq \text{min\_overlap} \) then \( \text{min\_overlap} \leftarrow \text{max\_rel\_ov} \)

\( \text{ret\_k} \leftarrow k \)

Return \( \text{ret\_k} \)

It can be shown that algorithm s-max is optimal for the sum-objective and the max-objective in problems where \( M \leq |\mathcal{T}| \) and \( n < |\mathcal{T}| \). It is also optimal for the max-objective if \( |\mathcal{T}| \leq M \leq 2|\mathcal{T}| \) or all agents request data of the same size.

It is observed that the relative performance of algorithm and main conclusion do not change. If \( p \) approaches to 0, it becomes easier to find guilty agents and algorithm performance converges. On the other hand, if \( p \) approaches 1, the relative differences among algorithms grow since more evidence is need to find an agent guilty.

The algorithm presented implements a variety of data distribution strategies that can improve the distributor’s chances of identifying a leak. It is shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in the data that agents must receive.

IV. CONCLUSION

However, in many cases we must indeed work with agents that may not be 100% trusted, and we may not be certain if a leaked object came from an agent or from some other source. In spite of these difficulties, we have shown it is possible to assess the likelihood that an agent is responsible for a leak, based on the overlap of his data with the leaked data and the data of other agents, and based on the probability that objects can be “guessed” by other means. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor’s chances of identifying a leak.

ACKNOWLEDGMENT

Students work is incomplete until they thank the almighty & his teachers. I sincerely believe in this and would like to thank Ms.M.SreeDevi, M.Tech(P.H.D), Associate prof., of the department of Computer Science & Engineering, MITS, Madanapalle, for her encouragement, motivation and for guiding me to write this paper. Also I am grateful to Dr.E.Madhusudhana Reddy, Prof., in CSE, and Head Of the Department of Computer Science, MITS, Madanapalle, Chittoor(dt), Andra Pradesh.

REFERENCES


International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati


Multiple Find and Replace Tool for Text Editors

Vignesh Prabu, Sivagnanam Mutharasu & Ramajayam
Department of Computer Science, Sri Krishna College of Engineering and Technology Coimbatore, India
E-mail : vigneshprabuu@gmail.com, skannanmunna@yahoo.com, ramajayam.g09@gmail.com

Abstract – In text editors like Notepad and vi editor, we can find a string and replace it with another one. It is not possible to find multiple strings and replace with a single string. This paper reveals other 2 approaches: Finding multiple strings and replacing it with a single string, finding single string and replacing it with multiple strings. Java based string manipulations have been carried out on the contents of the file to implement our functionalities. Thus, our tool provides excellent portability option.

Keywords - find and replace, multiple find, multiple replace, text editors, java swings.

I. INTRODUCTION

Find and replace itself has a rich legacy, dating back to teletype text editors such as Unix ed, in which the substitute command was the primary way to change existing text. Although the importance of find and replace has faded somewhat with the rise of direct-manipulation text editing, virtually every text editor still includes a find and replace command [3].

In today’s world, text editors have become standard medium of storing data. In academic, governmental and commercial environments, the majority of documents are now being prepared using Text Editors.

When repetitive changes must be made to a text file, there are several approaches to consider. The changes can be performed manually, which is tedious if there are many modifications to be made. Custom programs can be written to perform these changes automatically, but this requires programming skill, and familiarity with either the editor’s programming interface or file format [4]. Find and replace is one of the familiar methods to make repetitive changes.

This command in a typical text editor allows the users to choose one of the two alternatives: replace one match at a time or replace all matches at once.

If we want to change n number of different strings by a single string, then we will use the find and replace command n times. This is slow and difficult. Eventually, the user starts getting bored to implement the same operation n times.

Also, if we want to replace data-x with data-1 in its first occurrence, data-2 in its second occurrence, …, data-n in its nth occurrence, data-1 in its (n+1)th occurrence, data-2 in its (n+2)th occurrence and so on then we have to do it individually for all the occurrences. But this method is tedious since the whole document has to be read manually to find all the positions of the string data-x to finish this job.

This paper explores new alternatives to find and replace patterns in a text file. This approach provides the way to search multiple patterns in a file and replace it with a single pattern and, also to search single pattern in the file and replace it with multiple patterns in sequential manner. In this way repetitive changes can be done in a single operation rather than individually capturing the patterns and replacing it. It will be very useful in editing newspapers, scientific documents, etc. where search operations are extensively used.

The rest of this paper describes our experience with designing and evaluating user interface for multiple find and replace. After surveying related works, we present the user interface, highlighting our implementation results and pilot evaluations. Then we present details of our algorithm used. Finally, we discuss some of the conclusions that can be drawn from our experience.

II. RELATED WORKS

There are various algorithms proposed to perform the task find and replace. Some of them are Cluster-Based Find and Replace, Advanced Find and Replace, HandyFile Find and Replace, HTML Search and Replace etc. [3]. Cluster Find and Replace perform conventional find and replace based on clusters i.e. on parts of files [3]. Similarly, HandyFile Find and Replace provides the same functionality on multiple files [2]. Structural find and replace is used extensively on java source codes. It is used to search for instances of the classes, expressions etc and replace it [7]. Find and replace method implemented by new operational
transformation algorithm, provides string operations based find and replace [8].

But all these algorithms search a single string and replace one of its occurrence or all of its occurrences by another string.

Consider a scenario in which we have to replace multiple patterns by a single pattern. For example, if we want to replace the entries such as “1000”, “2000”, “3000” in a file with a single pattern “222”. What can we do? We have to perform the usual find and replace 3 times. Will it be possible to do the same for 100 different patterns? Yes, it is, but we need to perform the find and replace command 100 times. But our new method will achieve this in a single stroke.

Consider another scenario. We want to replace the pattern “ram sundar” with “ram” in its first occurrence, “sundar” in its second occurrence, “ram” in the third occurrence, “sundar” in the fourth occurrence and so on. Is it possible? Yes it is, but it takes n times to perform find and replace operations where n is the number of occurrences of “ram sundar”. In this case, we need to make the changes manually i.e. we have to change the ith occurrence of “ram sundar” by “ram” when ‘i’ is odd and the ith occurrence of “ram sundar” by “sundar” when ‘i’ is even. This way of approach is very slow and complicated. User concentration is required for every replacement, which doesn’t scale to large, complicated task. Eventually, the user starts getting bored and impatient to perform find and replace for each pattern, which leads to error prone situations. This functionality can be achieved using our modified find and replace method in an easy manner.

One of the major advantages of our method is that the replacement operations are carried on character level and hence any kind of pattern can be found and replaced. It can be used on program codes since it has the capability of recognizing numbers and special characters in addition to the alphabets.

The only character that is not recognized in our method is “$” since it is used as a delimiter to separate multiple patterns.

III. PROPOSED METHODOLOGY

A. Multiple find

In this section, we propose an algorithm to spot multiple patterns and replace it with a single pattern. The algorithm takes as input a set of patterns (which are subtrings of a document) which has to be replaced and the replacement pattern. Our algorithm has the following steps.

1) Break the replacement pattern into a character array [1].

2) Break the first pattern to be replaced into a character array.

3) Read the lines in the file one by one and create an individual character array for each line.

4) Then extract the characters of the first pattern to be found.

   a) Compare those with the character array of each line and if the comparison extends to be equal to the length of the pattern to be found, a match is said to have occurred.

5) After locating the first match, replacement is done character by character of the replacement array onto the matching pattern iteratively. In this way, all the consecutive patterns are found and replaced.

6) Finally, the character array of the lines are concatenated into lines and written back into the file without any change in the indentation and format after flushing it. Similarly, character array of all the lines are concatenated sequentially and written onto the same file.

B. Multiple Replace

Our algorithm for replacing single pattern with sequential multiple patterns takes a series of inputs such as pattern to be found, multiple replacement patterns.

1) Read the lines in the file one by one and create an individual character array for each line.

2) Then extract the characters of the pattern to be found in an array.

   a. Compare those with the character array of each line and if the comparison extends to be equal to the length of the pattern to be found, a match is said to have occurred.

3) After finding the first occurrence of the match, first replacement string is broken down into a character array and the matching pattern is replaced with this replacement pattern appropriately.

   a. In this way the following consecutive matches are located and the consecutive replacement pattern is broken down into a character array and replacement takes place.

   b. The replacement patterns count is iterated using a modulo function so that found matches gets replaced by the consecutive replacement patterns even if the number of occurrences of the match is greater than the number of replacement pattern.
4) Finally, the broken down array of characters of the line are concatenated into a line and written onto the same file. Similar procedure is carried out for all the lines in the file. Before doing this, the file is flushed so that no overwriting takes place.

One of the significant merit of our algorithm is that, it has the ability to adjust the space needed i.e. if the length of the pattern to be found is greater than the replacement pattern unnecessary spaces in the array of that line are removed during replacement or if the replacement pattern is greater than the pattern to be found, the space needed are allocated automatically for replacement.

IV. IMPLEMENTATION RESULTS

This section describes user interface of our find and replace tool. The interface was created using Java Swings [5] [6].

A. Multiple Find

In this section, we implement our method of finding multiple strings and replacing with a single string.

Consider the text file “santa.txt” in which the words such as “Black Peter”, “Nikolaus” and “Saint Papa Noel” has to be replaced by “Santa Claus”.

![Fig. 1: Contents of “santa.txt” before the execution of multiple find command.](image)

Fig. 1 shows the content of the file “santa.txt” in which the strings to be found are highlighted by underline.

TABLE-I.

<table>
<thead>
<tr>
<th>Label</th>
<th>Strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strings to be Found</td>
<td>Nikolaus$Saint Papa Noel$Black Peter</td>
</tr>
<tr>
<td>Replacement String</td>
<td>Santa Claus</td>
</tr>
</tbody>
</table>

TABLE-II.

<table>
<thead>
<tr>
<th>String to be found</th>
<th>Number of occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nikolaus</td>
<td>1</td>
</tr>
<tr>
<td>Saint Papa Noel</td>
<td>2</td>
</tr>
<tr>
<td>Black Peter</td>
<td>3</td>
</tr>
</tbody>
</table>

TABLE-I provides the details about the strings to be found and the string to be replaced whereas TABLE-II provides the number of occurrences of each pattern to be found.

Fig. 2 shows the multiple find and replace tool in action. Browse button in figure 2 is used to choose the file to be operated on (“santa.txt”). Also we can select the options whether to operate on partial words or whole words by selecting the appropriate radio buttons.

Fig. 3 shows the contents of the file after the execution of multiple find command. We can see that the strings “Nikolaus”, “Saint Papa Noel” and “Black Peter” have been replaced by the string “Santa Claus”.

B. Multiple Replace

Consider the updated file “santa.txt” to implement multiple replace method. Figure 3 shows the contents of the updated file.

![Fig. 2 : Multiple Find and Replace Interface](image)

![Fig. 3 : Contents of “santa.txt” after the execution of multiple find command.](image)

TABLE-III.

<table>
<thead>
<tr>
<th>Label</th>
<th>Strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>String to be found</td>
<td>Santa Claus</td>
</tr>
<tr>
<td>Replacement strings</td>
<td>Black Peter$Nikolaus$Saint Papa Noel</td>
</tr>
</tbody>
</table>

TABLE-III provides the details about the strings to be found and the string to be replaced whereas TABLE-I provides the number of occurrences of each pattern to be found.
TABLE-IV.

<table>
<thead>
<tr>
<th>String to be found</th>
<th>Number of occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Santa Claus</td>
<td>6</td>
</tr>
</tbody>
</table>

TABLE-III provides the information about the string to be found and the replacement strings that are replaced in sequential order. TABLE-IV provides the number of occurrences of the searched string.

Fig. 4 shows the contents of the file after execution of the multiple replace command. We can see that the 3 occurrences of the string “Santa Claus” has been replaced to “Black Peter”, “Nikolaus” and “Saint Papa Noel” in the order in which it is given.

V. WORK FLOW DIAGRAM

VI. FUTURE DISCUSSION

As discussed earlier, our multiple find and replace method differs from conventional find and replace methods. But it cannot perform multiple find and multiple replace simultaneously i.e. it cannot find multiple patterns and replace them with multiple patterns. We are working on that which takes as input the pattern to be found and the replacement pattern in pairs so that multiple patterns can be found and replaced.

VII. CONCLUSION

Unlike the traditional replace-all and replace-with-confirmation approaches, multiple find and replace allows users to focus on different strings to be found and replaced.

We compared multiple find and replace with traditional find and replace in a small user study. And most of the users prefer to use our multiple find and replace over the conventional one because of its efficient search and replacement mechanism.

Multiple find and replace suggest ways that search and replace operation are carried in an efficient way. We hope that these techniques will evolve to reduce tedium and increase correctness in difficult tasks that demand human attention.

REFERENCES

[3] Robert C. Miller, Cluster-Based Find and Replace, MIT Computer Science and AI Lab Cambridge, USA.
Designing Secure Systems using AORDD Methodologies

In UML System Models

R. Ushasree & P. Raja Rajeswari
Madanapalle Institute Technology and Sciences, Madanapalle

Abstract – We propose a AORDD methodology, based on Aspect-Oriented Modeling (AOM), for incorporating security mechanisms in an application. The functionality of the application is described using the primary model and the attacks are specified using aspects. The security mechanism, modeled as security aspect, is composed with the primary model to obtain the security treated model. We illustrate how this can be done and show how the resulting system can be evaluated to give assurance that it is resilient to the given attack. In this paper we describe an aspect-oriented modeling (AOM) approach that eases the task of exploring alternative ways of addressing concerns during software modeling.

Keywords - Alloy, Aspect-oriented modeling, secure systems design, Security analysis

I. INTRODUCTION

In the commercial world, designing secure applications is impacted by various parameters, such as time-to-market, cost and effort involved. We propose a risk driven development approach for designing such applications.

For example, a Role Based Access Control (RBAC) model can be used to describe a solution to the banking system’s access control concern. A decision to address a concern in a particular manner can give rise to other concerns. For example, the RBAC solution to the access control problem gives rise to new concerns pertaining to the management of roles and permissions. In risk-driven development (RDD) security risks are identified, evaluated, and treated as an integrated part of the development.

II. BACKGROUND

2.1 ALLOY

we show how to formally verify that a security mechanism incorporated into a system is effective in protecting against a given security breach. we show how a system modeled using UML is converted to a form that can be automatically verified using the Alloy Analyzer.the Alloy analyzer translates a model into Boolean expression and analyses it using SAT-SOLVERS metamodel element

2.2 ASPECT ORIENTED MODELING

An aspect oriented modeling approach of the following artifacts

1. A primary model that describes the business logic of the application.
2. A set of generic aspect models, where each model is a generic description of a crosscutting feature.
3. A set of bindings that determine where in the primary model the aspect models are to be composed.
4. A set of composition directives that influence how Aspect models are composed with the primary model.

2.3 SECURE SYSTEM DESIGNS

We also include the project-specific consequence of incorporating a security mechanism to prevent the attack, in the form of variables related to the development effort in terms of cost and time.

2.4 SECURITY ANALYSIS

In security assessment and management several techniques for identifying and assessing security problems in an information system are combined into a process that ensures that there is continuous review and update of its security controls.

• Eavesdropping - The attacker may observe the communications channel.
Designing Secure Systems using AORDD Methodologies In UML System Models

- **Replay** - The attacker records messages she has observed and re-sends them at a later time.
- **Man-in-the-middle** - The attacker intercepts the messages sent between the parties C and S and replaces these with her own messages.

### III. CASE STUDY

**EXAMPLE E-COMMERCE SYSTEM**

Our example is an e-commerce platform called ACTIVE. ACTIVE provides services for electronic purchasing of goods over the Internet. The project identified several security risks, including attacks against user authentication in the login service. Here we defines two models are primary model and context-model primary model that describes a user management system in which the UserMgmt class defines operations for adding a user to the repository (addUser) and for deleting a user from the repository (deleteUser), the diagrams of primary and context model.

![Fig. 1](image)

misuse model of original(a) ACTIVE login sequence and MIM attack, created by primary login sequence model with context-specific MIM passive attacks models. The communication between ACTIVE CLIENT and(B) LOGIN MANAGER through ATTACKER.

The attack is successful is ©ATTACKER obtains home page, or username,password.

#### 3.1. THE MAN-IN-THE-MIDDLE ATTACK

In this section, we show how to represent the man-in-the-middle attack as a generic aspect. Messages between a requestor and authenticator are intercepted by an attacker, authenticator.

#### 3.2. SECURITY MECHANISMS To COUNTER MAN-IN-THE-MIDDLE ATTACKS

System designers must identify security properties relevant to mitigating a risk to system assets. We identify properties according to the ISO/IEC TR 13335:2001 Information Technology—Guidelines for Management of IT Security [9].

#### 3.3. MISUSE MODEL OF SECURITY-TREATED PRIMARY MODEL

The SRP security-treated misuse model. However, the active attack differs in three ways:

1. Attacker substitutes its own expression and name in the startComm message (aExpr and a name)
2. Attacker generates its own key and token (key and aTok)
3. Attacker substitutes its token for the ActiveClient in the verify message (aTok).

![Fig. 2](image)
IV. IMPLEMENTATION

A constraint is a requirement which leaves no design option. e.g. the developers could use any language they like then say so. Otherwise describe just the constraint. When referring to system interfaces, legacy systems and databases refer to the design documentation for these. Add important diagrams to Appendix A and refer to them in the text. If there is insufficient information about these external systems then mention that this information will need to be completed for the purposes of the development of this system.

V. RESULTS AND DISCUSSIONS

Solutions to design concerns (e.g., security and fault tolerance concerns) may crosscut many modules of a design model. The cross-cutting nature of these solutions can make understanding, analyzing and changing the solutions difficult. This complexity can be addressed through the use of aspect-oriented modeling (AOM) techniques, where the design of a cross-cutting solution is undertaken in an independent fashion, and the resulting aspect models are composed with primary models of core functionality to create a complete system design. Composition is necessary to identify conflicts across aspect and primary models, and to identify undesirable emergent properties in composed models.

Example 1. Consider the example in Figure . In the context specific aspect model, the UserMgmt class contains an operation called getRepositorySize() that retrieves the size of SystemMgmtAuthRepository. a different operation. To resolve this conflict, the rename directive can rename one or both operations, and the replaceReferences directive can update any references to the old Name. The following composition directives are applied:

1. **rename** aspect::UserMgmt::getRepositorySize() to aspect::UserMgmt::getAuthRepositorySize()

2. **replaceReferences**

   aspect::UserMgmt::getRepositorySize() with aspect::UserMgmt::getAuthRepositorySize()

3. **renameprimary**::UserMgmt::getRepositorySize() primary::UserMgmt::getUserRepositorySize()

4. **replaceReferencesprimary**::UserMgmt::getRepositorySize() primary::UserMgmt::getUserRepositorySize() after Application and note the changes

The result of applying the directives is shown in Figure 3. Where applicable, the effects of the composition directives are denoted in the composed model using the corresponding numbers. The names of getRepositorySize() in aspect and primary are changed to getAuthRepositorySize() and getUserRepositorySize(), respectively. The references to the operation names are changed throughout each model to reflect the name change, and to avoid reference conflicts.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a methodology for developing secure systems that are resilient to given attacks. We first perform risk assessments to identify the types of attacks that are typical for such applications. We show how to evaluate the application against such attacks. If the results of this evaluation indicate that the assets may be compromised, then some security mechanism must be incorporated into the application. The resulting system is then formally analyzed to ensure that it is indeed resilient to the given attack. We validated our approach on a real-world e-commerce
application. Our approach does not detect new vulnerabilities but it can be used for assessing whether a given vulnerability poses sufficient risk that necessitates its mitigation. The main benefit of our approach is that it simplifies the design of complex systems. The primary models and the aspects can be analyzed in isolation to ensure that individually they satisfy the functional and security properties respectively. The models can be composed and the analysis of the composed model will give assurance that the resulting system also satisfies the properties. Another benefit of our approach is that it allows one to experiment with various security mechanisms to see which one is most suitable for preventing a given attack on the application. When a system is required to enforce different security properties, multiple aspects must be integrated with the application. This will allow one to study and formalize the interaction between aspects.

Our on-going and future work concentrates efforts in three areas. We are in the process of developing detailed algorithms to support the abstraction of complex UML diagrams and their conversion to OCL specifications, so that the approach can be automated. This ability will aid developers using the approach by reducing the chances that simplifying abstractions made by the developer leave out crucial items for the analysis. We are also investigating the broader applicability of the approach to other security mechanisms that are more appropriately specified by UML diagrams other than sequence diagrams. Finally, we are also investigating application of the approach to other stages in the development lifecycle of complex software systems, especially to the requirements phase.

REFERENCES


A Case Study on Storage Area Network (SAN)

Sanket Bengali
School of Information Technology, Advanced Software Technologies,
International Institute of Information Technology, Pune, Maharashtra, India
E-mail: sanketbengali.23@gmail.com

Abstract – For managing large amount of data, many companies are providing products for “Storage virtualization.” It is the way of managing the storage most efficiently and providing the customers high data availability. EMC2, Netapp, IBM etc. are some of the big players in this field. Storage Area Network (SAN) is the technology for providing virtualization of the storage to the servers in order to store and manage customers’ data and serve customers’ I/O requests in a better way. This paper gives an overview on architecture of SAN and a brief introduction about the components used in it. The paper also describes the way I/O request is handled with these components and the communication path between them.

Keywords - Storage Area Network, Fibre Channel over Ethernet, Host Bus Adapter, Converged Network Adapter, Redundant Array of Independent Disks, Logical Unit Number, SAN Boot

I. INTRODUCTION

The objective of this paper is to discuss the architecture of SAN with different components used in it and how different components communicate with each other and serve customers’ I/O requests. Different components like servers, adapters, switches, controllers, storage are used to build SAN which is used to provide high data-availability and efficient management of storage.

SAN includes hosts or servers which provide services for I/O of customers' data which come through any form of applications, switches for handling special storage protocols (like FC, iSCSI, FCoE), HBAs or CNAs (according to the protocol used), the controller for providing virtualization of the disks to the host and finally the disk controller like RAID. SAN provides RAS advantages – Redundancy, Availability and Serviceability.

II. ARCHITECTURE OF SAN

A. Servers

As shown in the figure, the servers or the hosts can be considered as the starting point of the I/O process through SAN.

From any application like a web-application or a windows-application which needs to deal with large amount of data, sends request for I/O of data to these hosts and the hosts need to serve these requests.
These hosts can be from multiple vendors like IBM servers, SUN servers, HP servers etc. With different platforms like Linux (all flavors), windows, sun Solaris, VMware, HP UNIX etc.. In this way, “heterogeneous environment” is preserved, that is the “storage virtualization” isn't dependent on any particular platform. Only some configuration may vary platform-to-platform.

B. Switch and adapter

From the host, the data goes to switch. This switch is required as multiple hosts need to be connected to one or more storage controllers. So, this switch contains number of ports which connects the hosts to the controllers.

On the host and controller side, special adapters are needed to communicate through the switch. These adapters are called as Host Bus Adapter (HBA) in case of FC protocol or Converged Network Adapter (CNA) in case of FCoE/iSCSI protocol. In short, these adapters are “protocol handlers” which transport SCSI data from host to controller through switch and vice versa. They may have single or double ports. The adapters on the host side works as “an initiator” (which initiates a command) and on the controller side, it works as “a target” (which executes the command) to the host and “an initiator” to the disk. So, an adapter can work as an initiator or a target or both initiator and target.

These adapters contain Small Form-factor Pluggable (SFP) transceivers which interface a network device like switch, adapter, router etc.. to a fiber optic or copper networking cable.

Protocols used in SAN

Here it is important to mention that FC, FCoE and iSCSI are the protocols which are used for data communication in SAN. Each protocol has its own advantages and disadvantages. FC protocol provides 2/4/8 GBps of data transfer speed, FCoE provides 10 GBps speed and iSCSI provides 1 GBps and 10 GBps data transfer speed.

Multi-pathing

Multiple switches can also be used which brings a concept called “multi-pathing” for data transfer. So, the hosts and the controllers can be connected to 2 switches such that they provide redundant paths for data transfer. As shown in the figure, the host and the controller have 2 paths.

Multi-pathing has mainly 2 uses: Data-availability and load-balancing.

Data availability comes into picture when there is less data transfer and hence the data communication is done through only one path and if that path fails, the multi-pathing software will sense that I/O has taken too long, then resets the connections and passes the request over an alternate path and thus the data communication continues. Here the application won't know anything went wrong and so, the data availability to the customer is not affected in case of failure of one path. So, this type of connectivity is called “active/passive” as only one path remains active at a time and the others remain idle until the active path gets failed. Typically, midrange storage systems are implemented with this type of connectivity.

And load-balancing comes into picture when there is high amount of data transfer and hence the 2nd path balances the load of data. So, this type of connectivity is called “active/active” as multiple paths remain active to the storage and the servers simultaneously use all of them to get more storage bandwidth. The multi-pathing software for this type of connectivity is sold by SAN vendors as they are complicated to implement. High-end storage systems are implemented with this type of connectivity. [3]

In case of multi-pathing, the hosts should have multi-pathing drivers as per the platform and some configuration is required for a vendor specific controller. If multi-pathing drivers are not installed or disabled, then the host will see 2 separate copies of each disk mapped to it through the controller which may bring data-inconsistency or data-redundancy etc.. problems.

Zoning

There is one more concept called “zoning” which is used for logical segmentation of the hosts, nodes and disk arrays in the fabric into a group. So, zone is a set of members that have access to each other. A member can be in multiple zones and there can be multiple zones in a zoneset. Multiple zonesets may be defined in a fabric but only one can be active at a time.

Zoning can be categorized into 3 types:

Port zoning: It uses the FC addresses of the physical ports to define zones. This method is secure, but it requires updating of zoning configuration information in the event of fabric reconfiguration and hence it is also called as hard-zoning.

WWN zoning: It uses WWNs (World Wide Names) to define zones. It allows the SAN to be recabled without reconfiguring the zone information. This is possible because the WWN is static to the node port. It is called as soft-zoning.

Mixed zoning: It combines the qualities of both WWN zoning and port zoning. Using mixed zoning enables a specific port to be tied to the WWN of a node. [1]
Every switch contains one default zoneset and one default zone which allows every connected member to have access to each other. Users can create their own zones according to their requirements.

It is a good practice to create 3 zones: One contains hosts and controller as members, one contains controller and disk arrays as members and one contains only controller as members.

These switches can be from multiple vendors like Cisco, Brocade etc..

And the HBAs or CNAs can be from QLogic, Emulex, Brocade, Intel etc..

C. Storage controller

After switch, the data goes to the “controller”. This is the mastermind of the whole game. It works as both, “target” to the host and “initiator” for the disk arrays. The main function of the controller is to provide “storage virtualization” to the hosts. It gives the hosts a virtual view of the disks such that the hosts can access them as if they are locally attached to them. So, the hosts can access each and every block of the disks and perform IO operations on them. Here it is important to mention that the iSCSI, FC and FCoE all are block level protocols which are used to transfer the data into chunks of blocks. And the virtualization here is called “in-band” virtualization as every communication that happens between the hosts and the disk arrays passes through the controller.

The most important thing in building SAN is that there should not be single point of failure. Hence, high-end storage solutions are designed with large number of controllers and cache memory and midrange storage solutions are designed with two controllers. So, in case of failure of one controller, the other takes care of the IO requests from the hosts without interrupting the customer services. One or more of such controllers or nodes form a “cluster”.

Cache

Cache memory is used for high performance of the I/O requests as they are faster in access than physical disks. So, the data from the hosts comes to the cache and immediately the hosts are acknowledged and from the cache, the data goes to the controller to the disk arrays. These cache are large in size (up to several GBs) and they are implemented with intelligence like read-ahead for read operations and write-back or write-through for write operations. They can be dedicated for read and write operations or global. [1]

So, the controllers are implemented with the mechanisms of handling I/O requests between hosts and disk arrays, management of the cache, management of the cluster, handling fail-over etc..

Besides “storage virtualization”, the controllers from different vendors also provide some extra functionalities like disk-mirroring (backup copy of data at local or remote location), flash-copy (point-in-time image of disk for testing, data mining purpose ), space-efficient disks or thin-provisioning (showing the server more capacity of a virtual disk than its actual physical capacity) etc..

EMC’s CLARiiON (midrange) and Symmetrix (High-end), IBM’s SVC(SAN Volume Controller) and Storwize V7000, Netapp’s FAS series and V-series etc. are some of the famous examples of controllers.

D. Disk arrays

And finally comes the physical disk arrays where actual data is stored. They are implemented with RAID arrays. Most famous are RAID 0,1,5,6 and 10. These RAID arrays give the controller the virtual view of the physical disks and from this virtual view, the controllers give the hosts the virtual view of the physical disks such that the hosts see them as their own local partitions. Physical disks can be logically split into logical volumes known as LUNs (Logical Unit Numbers).

So, with host-LUN mapping, the hosts can perform I/O operations on LUNs mapped to them and provide customers the service of their requests/responses for data.

SAN Boot

Host-LUN mapping brings one more concept into picture called “SAN Boot”. In normal case, the hosts have their own hard-disks of several GBs on which required Operating System can be installed and from that hard-disk, the host can boot the OS and serve the requests. But, as with storage virtualization, LUNs can be mapped to hosts regardless of the Operating System, and even Operating Systems can be installed on the LUN and can be booted by the host to which that LUN is mapped. The only thing required is that the host should have HBA or CNA configured with supported firmware and drivers for SAN Boot and proper zoning is done in the fabric such that the host can see the mapped LUNs. So, before booting the Operating System from the local hard-disk, the host can see the LUNs and can boot the Operating System from the LUN itself without using local hard-disks. The HBA or CNA needs to be configured according to the vendor (Emulex, QLogic etc...) instructions for SAN Boot such that at the BIOS time, it detects the LUN mapped to the host and can boot the OS from that. Local hard-disks need to be disabled or disconnected from the host. One more point here need to be taken care of is, while installing the OS on the LUN, there should be only one path from the host.
to the storage. After the installation, other paths can be added for multi-pathing.

SAN Boot has many benefits like alleviation of necessity for each server to have its own direct-attached disk and hence eliminating potential internal disk failure, centralized management of disk images and hence upgrades and fixes can be managed at a central location, remote mirroring of disks and snapshots features of SAN simplify the disaster and server failure recovery, high availability as a typical data center is highly redundant in nature – redundant paths, redundant disks and redundant storage controllers, requires less space, less power etc.. [4]

III. CONCLUSION

SAN (Storage Area Network) technology has emerged to overcome the limitations of DAS (Direct Attached Storage) like scalability, high data-availability and efficient management of storage. Different components like servers, adapters, switches, controllers, storage are used to build SAN. Applications running on the server are the sources of data and these data need to be stored into or fetched from physical storage to serve customers’ request. In SAN from the server, data goes to the switch to the controller and then form the controller to the disk arrays for write of data and in the reverse order for read of data. Hence the data passes through multiple layers, all with redundant paths for fail-over. iSCSI, FC, FCoE are the block level protocols used for data transfer.

ACKNOWLEDGEMENT

I wish to express my gratitude and sincere thanks to all who guided me in understanding the technology.

REFERENCES

[1] Information Storage and Management, Edited by G. Somasundaram, Alok Shrivastava, EMC Education Services, Wiley Publishing Inc., Indianapolis, Indiana


Seclusion Location Based Queries In Mobile Environments

C.Vijaya Lakshmi & R. Raja Sekhar
Department of CSE, JNTUA College of Engineering, Anantapur, Andhrapradesh, India
E-mail : vijayaa.cs@gmail.com, drasharaj@gmail.com

Abstract – In Location based services users with location-aware mobile devices are able to make queries about their surroundings anywhere and at any time. Location cloaking is one typical approach to protecting user location privacy in location based services. Upon receiving a location-based spatial query from the user, the system cloaks the user current location into a cloaking region based on the user privacy requirements.

Existing system controls the generation of cloaking regions and designed two cloaking algorithms, namely MaxAccu_Cloak and MinComm_Cloak. MaxAccu_Cloak is designed to maximize the accuracy of query results and MinComm_Cloak attempts to reduce the network communication cost. Two query processing modes, namely bulk and progressive.

This paper presents extends the mobility aware location cloaking technique to t-closeness privacy metric. t-closeness requires that the distribution of a sensitive attribute in any equivalence class is close to the distribution of a sensitive attribute in the overall table. t-closeness protects against attribute disclosure but not identity disclosure.

Keywords - Location based services, location privacy, query processing, mobile computing, t-closeness.

I. INTRODUCTION

Location Based Services are emerging as a major application of mobile geospatial technologies. In LBS users are able to make queries about their surroundings anywhere at any time. Spatial range queries and k-nearest-neighbor queries are two types of the most commonly used queries in LBS.

Location cloaking is one typical approach to protecting user location privacy in LBS. Receiving a location based spatial query from the user the system cloaks the user's current location into a cloaking region based on the user's privacy requirement. The location based spatial query is transformed into a region-based spatial query before being sent to the LBS server. The LBS server evaluates the region-based query and returns a result superset, which contains the query results for all possible location points in the cloaking region. Finally, the system refines the result superset to generate the exact results for the query location.

Existing system two query processing algorithms bulk algorithm that generates the query results all at once at the end of query evaluation and a progressive algorithm that produces the results incrementally during query evaluation. To achieve identity anonymity in LBS by spatiotemporal cloaking based on k-anonymity model, that is, the cloaked location is made indistinguishable from the location information of at least k-1 other users. To perform spatial cloaking used Quad-tree-like algorithm.

In this paper we present t-closeness privacy metric. We propose a novel privacy notation called t-closeness that formalizes the idea of global background knowledge by requiring that the distribution of a sensitive attribute in any equivalence class is close to the distribution of attribute in the overall table. This effectively limits the amount of individual-specific information an observer can learn. Further, in order to incorporate distance between values of sensitive attributes, we use the Earth Mover Distance metric to measure the distance between the two distributions.

This paper is organized as follows: Section-2 k-anonymity for location privacy, Section-3 provides bulk and progressive query processing modes, Section-4 provides t-closeness A new privacy measure, Section-5 provides Conclusion and Results.

II. k-anonymity FOR LOCATION PRIVACY

In the context of location privacy, the k-anonymity metric was initially adapted to measure microscopic location-privacy by Gruteser and Grunwald [1]. In this model, each query sent to the LBS (including the user's pseudonym, her position and the query time) is equivalent to one entry in a database, and the location-time information in the query serves as the quasi-identifier. In order to protect a user's location privacy...
using $k$-anonymity, each of her queries must be indistinguishable from that of at least $k-1$ other users. To this end, the pseudonyms of these $k$ users are removed from their queries, and the location-time pair in their queries is obfuscated to the same location-area and time-window, large enough to contain the users’ actual locations. The $k$-anonymity scheme for location privacy has become very popular, mainly due to its simplicity. A large body of research has focused on increasing the efficiency of $k$-anonymity schemes and reducing their cost of query obfuscation [2,3,4,5], extending the obfuscation method to protect traces [2] (i.e., location privacy at the macroscopic level), or adapting the architecture presented in [1] to different scenarios [6,7]. All of these systems can be represented by the initial model introduced in [1].

There is a set of users who access a LBS through a trusted Central Anonymity Server (CAS). Users send their LBS queries $(i,q,l)$ to the CAS, where $i$ is the identity of the user, $q$ is her query, $l$ is her precise location (expressed as a point with coordinates $(x,y)$ in a 2-dimensional space), and $t$ is the time at which the query is generated. In order to protect users’ privacy, the CAS removes the identity $i$ of the users. Furthermore, it obfuscates the location $l = (x, y)$ and the time $t$ at which the queries were generated. For this, it constructs a cloaking region $R = \{(x_1, x_2), (y_1, y_2), (t_1, t_2)\}$ such that there are at least $k$ users in $R$ whose location $l = (x, y)$ at time $t$ satisfies that $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, and $t_1 \leq t \leq t_2$. We note that, decentralized approaches [15] can also be represented by our model, by considering that the cloaking region $R$ is computed by a set of entities (e.g., users them-selves) in a distributed manner (i.e., they collectively play the role of the CAS). Moreover, our model can accommodate both the systems in which users have to continuously report their location to the CAS [3], in order to build optimal regions, and the systems that rely only on user-triggered discrete queries for that purpose [2]. The $k$-anonymity location obfuscation technique aims at achieving two properties: query anonymity and location privacy. Achieving query anonymity implies that it is not possible for the adversary to link the identity $i$ of a user to her query $q$, based on the location information (cloaking region) associated with the query. Location privacy is achieved when it is not possible for the adversary to learn the location $l$ of a user $i$ at time $t$, using queries he receives from the users and his a priori knowledge.

We consider an adversary that controls the LBS and, in addition to the received queries, has access to some background information. For example, one of the threats considered in [1] is restricted space identification”. In this threat scenario the adversary knows that a given location corresponds (exclusively) to a user address, meaning that a query coming from that precise location would be linked to the user who resides at that address. Another considered threat in [1] is that the adversary (in addition to controlling the LBS) may deploy antennas in the vicinity of the users and thus knows that a given user is in location $l$ at time $t$. Limitation of $k$-anonymity in situations where the values of sensitive attributes are not diverse. We propose to overcome this limitation by t-closeness privacy metric.

### III BULK AND PROGRESSIVE QUERY PROCESSING MODES

Two query processing algorithms:

A bulk algorithm that generates the query results all at once at the end of query evaluation. The bulk query processing algorithm generates the kCRNN results at the end of query evaluation. The server cannot start transmitting the results to the client until end of query evaluation. Therefore, the query response time increased when we use bulk algorithm.

An alternative for this is progressive query processing algorithm to parallelize the query evaluation and result transmission. A progressive algorithm that produces the results incrementally during query evaluation. When compare to bulk progressive algorithm query response time is reduced. To reduce this query response time rapidly we propose t-closeness privacy metric it reduces query response time.

### IV. t-closeness: A NEW PRIVACY MEASURE

Privacy measured by the information gain of an observer. Before seeing the released table, the observer has some prior belief about the sensitive attribute value of an individual. After seeing the released table, the observer has a posterior belief. Information gain can be represented as the difference between the posterior belief and the prior belief. We separate the information gain into two parts: that about the whole population in the released data and that about specific individuals.

To motivate our approach, let us perform the following thought experiment: First an observer has some prior belief $B_0$ about an individual’s sensitive attribute. Then, in a hypothetical step, the observer is given a completely generalized version of the data table where all attributes in a quasi-identifier are removed (or, equivalently, generalized to the most general values). The observer’s belief is influenced by $Q$, the distribution of the sensitive attribute value in the whole table, and changes to $B_1$. Finally, the observer is given the released table. By knowing the quasi-identifier values of the individual, the observer is able to identify the equivalence class that the individual’s record and learn
the distribution $P$ of sensitive attribute values in this class. The observer's belief changes to $B_2$.

We choose to limit the difference between $B_1$ and $B_2$. In other words, we assume that $Q$, the distribution of the sensitive attribute in the overall population in the table, is public information. We do not limit the observer’s information gain about the populations as a whole, but limit the extent to which the observer can learn additional information about specific individuals.

To justify our assumption that $Q$ should be treated as public information, we observe that with generalizations, the most one can do is to generalize all quasi-identifier attributes to the most general value. Thus as long as a version of the data is to be released, a distribution $Q$ will be released. We also argue that if one wants to release the table at all, one intends to release the distribution $Q$ and this distribution is what makes data in this table useful. In other words, one wants $Q$ to be public information. A large change from $B_0$ to $B_1$ means that the data table contains a lot of new information, e.g., the new data table corrects some widely held belief that was wrong. In some sense, the larger the difference between $B_0$ and $B_1$ is, the more valuable the data is. Since the knowledge gain between $B_0$ and $B_1$ is about the whole population, we do not limit this gain. We limit the gain from $B_1$ to $B_2$ by limiting the distance between $P$ and $Q$. Intuitively, if $P = Q$, then $B_1$ and $B_2$ should be the same. If $P$ and $Q$ are close, then $B_1$ and $B_2$ should be close as well, even if $B_0$ may be very different from both $B_1$ and $B_2$.

Requiring that $P$ and $Q$ to be close would also limit the amount of useful information that is released, as it limits information about the correlation between quasi-identifier attributes and sensitive attributes. However, this is precisely what one needs to limit. If an observer gets too clear a picture of this correlation, then attribute disclosure occurs. The $t$ parameter in $t$-closeness enables one to trade off between utility and privacy. Now the problem is to measure the distance between two probabilistic distributions. There are a number of ways to define the distance between them. Given two distributions

$$ P = (p_1, p_2, ..., p_m), Q = (q_1, q_2, ..., q_m), $$

the variational distance is defined as:

$$ D[P, Q] = \sum_{i=1}^{m} |p_i - q_i|. $$

And the Kullback-Leibler (KL) distance [10] is defined as:

$$ D[P, Q] = \sum_{i=1}^{m} p_i \log \frac{p_i}{q_i} = H(P) - H(P, Q) $$

Where $H(P) = \sum_{i=1}^{m} p_i \log p_i$ is the entropy of $P$ and $H(P, Q) = \sum_{i=1}^{m} p_i \log q_i$ is the cross-entropy of $P$ and $Q$.

This $t$-closeness privacy metric reduce query response time when compare to bulk and progressive query processing algorithms.

### V. CONCLUSION AND RESULTS

In this paper we discussed k-anonymity for location privacy and have some limitations to overcome this we proposed new privacy metric is $t$-closeness. Below Table 1 shows the results of query response time compared to bulk, progressive and $t$-closeness.

<table>
<thead>
<tr>
<th>Query processing modes</th>
<th>Response Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk</td>
<td>57 46 35 30</td>
</tr>
<tr>
<td>Progressive</td>
<td>36 27 20 18</td>
</tr>
<tr>
<td>t-closeness</td>
<td>22 18 15 10</td>
</tr>
</tbody>
</table>

Table 1: The comparision table of response time of bulk, progressive and $t$-closeness.

### REFERENCES


Data mining Feature Clustering Algorithm in Text Classification

A.Yaganteeswarudu, J.Uma Mahesh & S.Lalitha
SJ CET, Kurnool, India
E-mail : yaganteeswar.a@gmail.com, umamaheshsjcet@gmail.com, lalitha_sunny@yahoo.com

Abstract – Feature clustering is a powerful method to reduce the dimensionality of feature vectors for text classification. In this paper, we propose a fuzzy similarity-based self-constructing algorithm for feature clustering. The words in the feature vector of a document set are grouped into clusters, based on similarity test. Words that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature, corresponding to a cluster, is a weighted combination of the words contained in the cluster. By this algorithm, the derived membership functions match closely with and describe properly the real distribution of the training data. Besides, the user need not specify the number of extracted features in advance, and trial-and-error for determining the appropriate number of extracted features can then be avoided. Experimental results show that our method can run faster and obtain better extracted features than other methods.

Keywords - Fuzzy similarity, feature clustering, feature extraction, feature reduction, text classification.

I. INTRODUCTION

In text classification, the dimensionality of the feature vector is usually huge. For example, 20 Newsgroups and Reuters21578 top 10, which are two real-world data sets, both have more than 15,000 features. Such high dimensionality can be a severe obstacle for classification algorithms. To alleviate this difficulty, feature reduction approaches are applied before document classification tasks are performed. Two major approaches, feature selection and feature extraction have been proposed for feature reduction. In general, feature extraction approaches are more effective than feature selection techniques, but are more computationally expensive. Therefore, developing scalable and efficient feature extraction algorithms is highly demanded for dealing.

II. BACKGROUND AND RELATED WORK

A. Purpose

We propose a fuzzy similarity-based self-constructing feature clustering algorithm, which is an incremental feature clustering approach to reduce the number of features for the text classification task. The words in the feature vector of a document set are represented as distributions, and processed one after another. Words that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. If a word is not similar to any existing cluster, a new cluster is created for this word. Similarity between a word and a cluster is defined by considering both the mean and the variance of the cluster. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature corresponding to a cluster is a weighted combination of the words contained in the cluster.

B. Feature Extraction

Word patterns have been grouped into clusters, and words in the feature vector W are also clustered accordingly. For one cluster, we have one extracted feature. Since we have k clusters, we have k extracted features. The elements of T are derived based on the obtained clusters, and feature extraction will be done. We propose three weighting approaches: hard, soft, and mixed. In the hard-weighting approach, each word is only allowed to belong to a cluster, and so it only contributes to a new extracted feature. In the soft-weighting approach, each word is allowed to contribute to all new extracted features, with the degrees depending on the values of the membership functions. The mixed-weighting approach is a combination of the hard-weighting approach and the soft-weighting approach.
C. Feature Reduction

In general, there are two ways of doing feature reduction, feature selection, and feature extraction. By feature selection approaches, a new feature set \( W' = \{w'_1, w'_2, \ldots, w'_k \} \) is obtained, which is a subset of the original feature set \( W \). Then \( W' \) is used as inputs for classification tasks. Information Gain (IG) is frequently employed in the feature selection approach. It measures the reduced uncertainty by an information-theoretic measure and gives each word a weight. The weight of a word \( w_j \) is calculated as follows:

\[
P(\text{cl}|w_j) = \frac{P(w_j|\text{cl})}{P(w_j)P(\text{cl})} \]

where \( P(\text{cl}) \) denotes the prior probability for class \( \text{cl} \), \( P(w_j) \) denotes the prior probability for feature \( w_j \), \( P(w_j) = 1 - P(w_j) \), and \( P(\text{cl}|w_j) \) and \( P(\text{cl}|\neg w_j) \) denote the probability for class \( \text{cl} \) with the presence and absence, respectively, of \( w_j \). The words of top \( k \) weights in \( W \) are selected as the features in \( W' \). 

IOC tries to find an optimal transformation matrix \( F^* \in \mathbb{R}^{m \times k^*} \), where \( k^* \) is the desired number of extracted features, according to the following criterion:

\[
F^* = \arg \max \text{ trace}(F^T S_b F),
\]

where \( F \in \mathbb{R}^{m \times k^*} \) and \( F^T F = I \), and

\[
S_b = \sum_{q=1}^p P(c_q)(M_q - M_{all})(M_q - M_{all})^T
\]

with \( P(c_q) \) being the prior probability for a pattern belonging to class \( c_q \), \( M_q \) being the mean vector of class \( c_q \), and \( M_{all} \) being the mean vector of all patterns.

D. Feature Clustering

Feature clustering is an efficient approach for feature reduction, which groups all features into some clusters, where features in a cluster are similar to each other. The feature clustering methods proposed in are “hard” clustering methods, where each word of the original features belongs to exactly one word cluster. The new feature set \( W' = \{w'_1, w'_2, \ldots, w'_k \} \) corresponds to a partition \( \{W_1, W_2, \ldots, W_k\} \) of the original feature set \( W \), i.e., \( W_1 \cap W_2 = \emptyset \), where \( 1 \leq q, t \leq k \) and \( q \neq t \). Note that a cluster corresponds to an element in the partition. Then, the feature value of the converted document \( d'_i \) is calculated as follows:

\[
d'_i = \sum_{w_j \in W_i} d_j,
\]

The distribution of a cluster \( W_t \) is calculated as follows:

\[
P(\text{cl}|W_t) = \sum_{w_j \in W_t} \frac{P(w_j)}{P(w_j)P(\text{cl})} P(\text{cl}|w_j).
\]

The goal of DC is to minimize the following objective function:

\[
\sum_{t=1}^k \sum_{w_j \in W_t} P(w_j)KL(P(C|w_j), P(C|W_t)),
\]

This takes the sum over all the \( k \) clusters, where \( k \) is specified by the user in advance.

III. OUR METHOD

There are some issues pertinent to most of the existing feature clustering methods. First, the parameter \( k \), indicating the desired number of extracted features, has to be specified in advance. This gives a burden to the user, since trial-and-error has to be done until the appropriate number of extracted features is found. Second, when calculating similarities, the variance of the underlying cluster is not considered. Intuitively, the distribution of the data in a cluster is an important factor in the calculation of similarity. Third, all words in a cluster have the same degree of contribution to the resulting extracted feature. Sometimes, it may be better if more similar words are allowed to have bigger degrees of contribution.

A. Self-Constructing Clustering

Our clustering algorithm is an incremental, self-constructing learning approach. Word patterns are considered one by one. The user does not need to have any idea about the number of clusters in advance. No clusters exist at the beginning, and clusters can be created if necessary. For each word pattern, the similarity of this word pattern to each existing cluster is calculated to decide whether it is combined into an existing cluster or a new cluster is created. Once a new cluster is created, the corresponding membership function should be initialized.

The clustering algorithm can be summarized below:

# of original word patterns: \( m \)  
# of classes: \( p \)  
Threshold: \( \rho \)  
Initial deviation: \( \sigma_0 \)  
Initial # of clusters: \( k = 0 \)  
Input:
\( x_i = <x_{i1}, x_{i2}, \ldots, x_{ip}> \), \( i < i < m \)
Output: 
Clusters G1, G2, . . ., Gk
procedure Self-Constructing-Clustering-Algorithm
for each word pattern xi, 1 <= i <= m
    temp_W = {Gj | µg(xi) >= ρ, 1 <= j <= k}
    if (temp_W == Φ)
        A new cluster Gh, h = k+1, is created
    else let Gt ∈ temp_W be the cluster to which xi is closest by
        t = arg max µg(xi)
        Incorporate xi into Gt by
            (mtj = (St x mtj + xij) / (St + 1)) - (St = St+1)
    endif;
endfor;
return with the created k clusters;
endprocedure

B. Feature extraction

Formally, feature extraction can be expressed in the following form:

\[ D' = DT, \]
\[ D = [d_1, d_2, ..., d_n]^T \]
\[ D' = [d_1', d_2', ..., d_n']^T \]

where

\[ T = \begin{bmatrix} t_{11} & ... & t_{1k} \\ \vdots & \ddots & \vdots \\ t_{mk} & ... & t_{mk} \end{bmatrix} \]

\[ d_i = [d_{i1}, d_{i2}, ..., d_{im}] \]
\[ d_i' = [d_{i1}', d_{i2}', ..., d_{im}'] \]

Clearly, T is a weighting matrix. The goal of feature reduction is achieved by finding an appropriate T such that k is smaller than m. In the divisive information theoretic feature clustering algorithm described in, the elements of T in are binary and can be defined as follows:

\[ t_{ij} = \begin{cases} 1, & \text{if } wi \in W_j; \\ 0, & \text{otherwise} \end{cases} \]

We propose three weighting approaches: hard, soft, and mixed. In the hard-weighting approach, in this case, the elements of T in are defined as follows:

\[ t_{ij} = \begin{cases} 1, & \text{if } \arg \max \mu_g(x_i) \\ 0, & \text{otherwise} \end{cases} \]

In the soft-weighting approach, the elements of T in are defined as follows:

\[ t_{ij} = \mu_{g_i}(x_i) \]

The mixed-weighting approach is a combination of the hard-weighting approach and the soft-weighting approach. For this case, the elements of T in are defined as follows:

\[ t_{ij} = (\gamma) x t_{ij}^H + (1 - \gamma) x t_{ij}^S \]

B. Text Classification

Given a set D of training documents, text classification can be done as follows: We specify the similarity threshold for, and apply our clustering algorithm. Assume that k clusters are obtained for the words in the feature vector W. Then we find the weighting matrix T and convert D to D0 by . Using D0 as training data, a classifier based on support vector machines (SVM) is built. Note that any classifying technique other Classification techniques have been applied to Spam filtering, a process which tries to discern E-mail spam messages from legitimate emails. Topic spotting, automatically determining the topic of a text Language identification, automatically determining the language of a text.

Automatic document classification

Automatic document classification tasks can be divided into three sorts: supervised document classification where some external mechanism (such as human feedback) provides information on the correct classification for documents, unsupervised document classification (also known as document clustering), where the classification must be done entirely without reference to external information, and semi-supervised document classification, where parts of the documents are labeled by the external mechanism than SVM can be applied. SVM is better than other methods for text categorization. SVM is a kernel method.

<table>
<thead>
<tr>
<th>TABLE I : A Simple Document Set D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>d_1</td>
</tr>
<tr>
<td>d_2</td>
</tr>
<tr>
<td>d_3</td>
</tr>
<tr>
<td>d_4</td>
</tr>
<tr>
<td>d_5</td>
</tr>
<tr>
<td>d_6</td>
</tr>
<tr>
<td>d_7</td>
</tr>
</tbody>
</table>
The objective function and constraints of the classification problem can be formulated as:
\[
\min \frac{1}{2} W^T W + C \sum_{i=1}^{l} \varepsilon_i
\]
where \(l\) is the number of training patterns, \(C\) is a parameter, which gives a tradeoff between maximum margin and classification error, and \(y_i\), being +1 or -1, is the target label of pattern \(x_i\).

An SVM described above can only separate apart two classes, \(y_i = +1\) and \(y_i = -1\). We follow the idea to construct an SVM-based classifier. For \(p\) classes, we create \(p\) SVMs, one SVM for each class. For the SVM of class \(c_v\), \(1 \leq v \leq p\), the training patterns of class \(c_v\) are treated as having \(y_i = +1\), and the training patterns of the other classes are treated as having \(y_i = -1\). The classifier is then the aggregation of these SVMs. Now we are ready for classifying unknown documents suppose, \(d\) is an unknown document. We first convert \(d\) to \(d'\) by \(d' = d^T\).

**V AN EXAMPLE**

We give an example here to illustrate how our method works. Let \(D\) be a simple document set, containing 9 documents \(d_1, d_2, \ldots, d_9\) of two classes \(c_1\) and \(c_2\), with 10 words “office,” “building,” . . . , “fridge” in the feature vector \(W\), as shown in Table 1. For simplicity, we denote the ten words as \(w_1, w_2; \ldots; w_{10}\), respectively. We calculate the ten word patterns \(x_1, x_2, \ldots, x_{10}\)

For example \(x_6 = \langle P(c_1/w_6), P(c_2/w_6)\rangle\) where \(P(c_2/w_6)\) is calculated as
\[
P(c_2/w_6) = 1 \times 0 + 2 \times 0 + 0 \times 0 + 1 \times 0 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 0 \times 0 + 1 = 0.50
\]

The resulting word patterns are shown in Table 2. Note that each word pattern is a two-dimensional vector, since there are two classes involved in \(D\).

### TABLE II

<table>
<thead>
<tr>
<th>Word Patterns of (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
</tbody>
</table>

The resulting word patterns are shown in Table 2. Note that each word pattern is a two-dimensional vector, since there are two classes involved in \(D\). We run our self-constructing clustering algorithm, by setting \(\sigma_0 = 0.5\) and \(\rho = 0.64\), on the word patterns. Obtain 3 clusters \(G_1, G_2, \) and \(G_3\), which are shown in Table 3. The fuzzy similarity of each word pattern to each cluster is shown in Table 4. The weighting matrices \(TH, TS, \) and \(TM\) obtained by hard-weighting, soft-weighting, and mixed weighting (with \(\gamma = 0.8\)) respectively.

### TABLE III

<table>
<thead>
<tr>
<th>Three Clusters Obtained</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
</tr>
<tr>
<td>(G_1)</td>
</tr>
<tr>
<td>(G_2)</td>
</tr>
<tr>
<td>(G_3)</td>
</tr>
</tbody>
</table>

### IV. EXPERIMENTAL RESULTS

In this section, we present experimental results to show the effectiveness of our fuzzy self-constructing feature clustering method. Three well-known data sets for text classification

#### A. Experiment 1: 20 Newsgroups Data Set:

The 20 Newsgroups collection contains about 20,000 articles taken from the Usenet newsgroups. These articles are evenly distributed over 20 classes, and each class has about 1,000 articles, as shown in Fig. 1a.

In this figure, the x-axis indicates the class number, and the y-axis indicates the fraction of the articles of each class.

### TABLE IV

<table>
<thead>
<tr>
<th>Fuzzy Similarities of Word Patterns to Three Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mu_{G_1}(x))</td>
</tr>
<tr>
<td>(\mu_{G_2}(x))</td>
</tr>
<tr>
<td>(\mu_{G_3}(x))</td>
</tr>
</tbody>
</table>
TABLE V
Micro averaged Accuracy (Percent) of Different Methods for 20 Newsgroups Data

<table>
<thead>
<tr>
<th>% of extracted features</th>
<th>20</th>
<th>58</th>
<th>84</th>
<th>120</th>
<th>203</th>
<th>280</th>
<th>521</th>
<th>1187</th>
<th>1453</th>
</tr>
</thead>
<tbody>
<tr>
<td>threshold (ρ)</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
<td>0.06</td>
<td>0.12</td>
<td>0.19</td>
<td>0.23</td>
<td>0.32</td>
<td>0.36</td>
</tr>
<tr>
<td>microaveraged accuracy</td>
<td>IG</td>
<td>95.79</td>
<td>96.20</td>
<td>96.32</td>
<td>96.40</td>
<td>96.81</td>
<td>96.94</td>
<td>97.33</td>
<td>97.86</td>
</tr>
<tr>
<td></td>
<td>DC</td>
<td>97.73</td>
<td>98.19</td>
<td>98.38</td>
<td>98.36</td>
<td>98.43</td>
<td>98.56</td>
<td>98.28</td>
<td>98.63</td>
</tr>
<tr>
<td></td>
<td>IOC</td>
<td>97.09</td>
<td>97.44</td>
<td>97.50</td>
<td>97.51</td>
<td>97.68</td>
<td>97.62</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H-FCC</td>
<td>97.94</td>
<td>98.43</td>
<td>98.44</td>
<td>98.41</td>
<td>98.63</td>
<td>98.59</td>
<td>98.65</td>
<td>98.61</td>
</tr>
<tr>
<td></td>
<td>S-FCC</td>
<td>98.46</td>
<td>98.57</td>
<td>98.62</td>
<td>98.59</td>
<td>98.64</td>
<td>98.69</td>
<td>98.69</td>
<td>98.70</td>
</tr>
<tr>
<td></td>
<td>M-FCC</td>
<td>98.10</td>
<td>98.54</td>
<td>98.57</td>
<td>98.58</td>
<td>98.56</td>
<td>98.60</td>
<td>98.59</td>
<td>98.60</td>
</tr>
</tbody>
</table>

Full features (25718 features): MicroAcc = 98.45

B. Experiment 2: Reuters Corpus Volume 1 (RCV1) Data Set

The RCV1 data set consists of 804,414 news stories produced by Reuters from 20 August 1996 to 19 August 1997. All news stories are in English, and have 109 distinct terms per document on average.

IOC when the number of extracted features exceeds 18. As shown in the figure, IG performs the worst in classification accuracy, especially when the number of extracted features is small.

Fig. 1. Class distributions of three data sets. (a) 20 Newsgroups. (b) RCV1. (c) Cade12

Fig. 2 Micro averaged accuracy (percent) of different methods for RCV1 data

TABLE VI
Micro averaged Accuracy (percent) of Different Methods for RCV1 Data

<table>
<thead>
<tr>
<th>% of extracted features</th>
<th>18</th>
<th>42</th>
<th>65</th>
<th>74</th>
<th>202</th>
<th>421</th>
<th>818</th>
<th>1700</th>
</tr>
</thead>
<tbody>
<tr>
<td>threshold (ρ)</td>
<td>0.01</td>
<td>0.03</td>
<td>0.07</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>microaveraged accuracy</td>
<td>IG</td>
<td>96.95</td>
<td>97.13</td>
<td>97.24</td>
<td>97.26</td>
<td>97.60</td>
<td>98.03</td>
<td>98.37</td>
</tr>
<tr>
<td></td>
<td>DC</td>
<td>98.03</td>
<td>98.07</td>
<td>98.19</td>
<td>98.29</td>
<td>98.39</td>
<td>98.51</td>
<td>98.58</td>
</tr>
<tr>
<td></td>
<td>IOC</td>
<td>96.98</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H-FCC</td>
<td>98.03</td>
<td>98.02</td>
<td>98.13</td>
<td>98.14</td>
<td>98.31</td>
<td>98.41</td>
<td>98.49</td>
</tr>
<tr>
<td></td>
<td>S-FCC</td>
<td>98.26</td>
<td>98.37</td>
<td>98.39</td>
<td>98.41</td>
<td>98.58</td>
<td>98.57</td>
<td>98.57</td>
</tr>
<tr>
<td></td>
<td>M-FCC</td>
<td>97.85</td>
<td>98.28</td>
<td>98.31</td>
<td>98.38</td>
<td>98.49</td>
<td>98.56</td>
<td>98.58</td>
</tr>
</tbody>
</table>

Full features (17152 features): MicroAcc = 98.83
C. Experiment 3: Cade12 Data

Cade12 is a set of classified Web pages extracted from the Cad Web directory [41]. This directory points to Brazilian Web pages that were classified by human experts into 12 classes. The Cade12 collection has a skewed distribution, and the three most popular classes represent more than 50 percent of all documents. A version of this data set.

VI. CONCLUSIONS

We have presented a fuzzy self-constructing feature clustering (FFC) algorithm, which is an incremental clustering approach to reduce the dimensionality of the features in text classification. Features that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. If a word is not similar to any existing cluster, a new cluster is created for this word. Similarity between a word and a cluster is defined by considering both the mean and the variance of the cluster. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature corresponding to a cluster is a weighted combination of the words contained in the cluster. By this algorithm, the derived membership functions match closely with and describe properly the real distribution of the training data. Besides, the user need not specify the number of extracted features in advance, and trial-and-error for determining the appropriate number of extracted features can then be avoided.

ACKNOWLEDGEMENT

We express our sincere thanks to SJCET providing us good lab facilities. A heart full and sincere gratitude to our beloved parents and friend for their tremendous motivation and moral support.

REFERENCES

Cloud Specific Issues and Vulnerabilities Solutions

B.Suresh Kumar, C.Kishor Kumar Reddy, Lokesh Naik & S.K. Prasanth  
Dept. of C.S.E, VCE, Samshabad, R.R (Dist), A.P, India  
E-mail : sureshkumargoud@gmail.com, Kishoar23@gmail.com, lokessh@yahoo.com, sk_p21@yahoo.co.in

Abstract – The current discourse about cloud computing security issues makes a well-founded assessment of cloud computing's security impact difficult for two primary reasons. First, as is true for many discussions about risk, basic vocabulary such as "risk," "threat," and "vulnerability" are often used as if they were interchangeable, without regard to their respective definitions. Second, not every issue that's raised is really specific to cloud computing. We can achieve an accurate understanding of the security issue "delta" that cloud computing really adds by analyzing how cloud computing influences each risk factor. One important factor concerns vulnerabilities: cloud computing makes certain well-understood vulnerabilities more significant and adds new vulnerabilities. Here, we define four indicators of cloud-specific vulnerabilities, introduce security-specific cloud reference architecture, and provide examples of cloud-specific vulnerabilities for each architectural component. This paper highlights and categorizes many of security issues introduced by the "cloud"; surveys the risks, threats and vulnerabilities, and makes the necessary recommendations that can help promote the benefits and mitigate the risks associated with Cloud Computing.

Keywords - cloud-specific vulnerabilities, risk, threat, delta.

VULNERABILITY: AN OVERVIEW

Vulnerability is a prominent factor of risk ISO 27005 defines risk as “the potential that a given threat will exploit Vulnerability of an asset or group of assets and thereby cause harm to the organization,” measuring it in terms of both the likelihood of an event and its consequence. The Open Group’s risk taxonomy offers a useful overview of risk factors (see Figure 1).

Fig. 1: Factors contributing to risk according to the open Group’s risk taxonomy.

- The frequency with which threat agents try to exploit vulnerability. This frequency is determined by both the agents’ motivation (What can they gain with an attack? How much effort does it take? What is the risk for the attackers?) and how much access ("contact") the agents have to the attack targets.
- The difference between the threat agents’ attack capabilities and the system’s strength to resist the attack.

This second factor brings us toward a useful definition of vulnerability.

DEFINING VULNERABILITY:

According to the Open Group’s risk taxonomy, Vulnerability is the probability that an asset will be unable to resist the actions of a threat agent. Vulnerability exists when there is a difference between the force being applied by the threat agent, and an object’s ability to resist that force. So, vulnerability must always be described in terms of resistance to a certain type of attack.

Vulnerabilities And Cloud Risk

We’ll now examine how cloud computing influences the risk factors in Figure 1, starting with the right-hand side of the risk factor tree. From a cloud customer perspective, the right-hand side dealing with probable magnitude of future loss isn’t changed at all by cloud computing: the consequences and ultimate cost of, say, a confidentiality breach, is exactly the same regardless of whether the data breach occurred within a cloud or a conventional IT infrastructure. For a cloud...
service provider, things look somewhat different: because cloud computing systems were previously separated on the same infrastructure, a loss event could entail a considerably larger impact. But this fact is easily grasped and incorporated into a risk assessment: no conceptual work for adapting impact analysis to cloud computing seems necessary. So, we must search for changes on Figure 1’s left-hand side—the loss event frequency. Cloud computing could change the probability of a harmful event’s occurrence. As we show later, cloud computing causes significant changes in the vulnerability factor. Of course, moving to a cloud infrastructure might change the attackers’ access level and motivation, as well as the effort and risk—a fact that must be considered as future work. But, for supporting a cloud-specific risk assessment, it seems most profitable to start by examining the exact nature of cloud-specific vulnerabilities.

Architectural Components of cloud computing

![Fig. 2: The cloud reference architecture.](image)

**ARCHITECTURAL COMPONENTS AND VULNERABILITIES:**

Cloud service models are commonly divided into SaaS, PaaS, and IaaS, and each model influences the vulnerabilities exhibited by a given cloud infrastructure. It’s helpful to add more structure to the service model stacks: Figure 2 shows a cloud reference architecture that makes the most important security-relevant cloud components explicit and provides an abstract overview of cloud computing for security issue analysis.

In addition to the original model, we’ve identified supporting functions relevant to services in several layers and added them to the model as vertical spans over several horizontal layers.

**OUR CLOUD REFERENCE ARCHITECTURE HAS THREE MAIN PARTS:**

**Supporting (IT) infrastructure:**

These are facilities and services common to any IT service, cloud or otherwise. We include them in the architecture because we want to provide the complete picture; a full treatment of IT security must account for a cloud service’s non-cloud-specific components.

**Cloud-specific infrastructure:**

These components constitute the heart of a cloud service; cloud-specific vulnerabilities and corresponding controls are typically mapped to these components.

**Cloud service consumer:**

Again, we include the cloud service customer in the reference architecture because it’s relevant to an all-encompassing security treatment.

Using the cloud reference architecture’s structure, we can now run through the architecture’s components and give examples of each component’s cloud-specific vulnerabilities.

**Cloud Software Infrastructure and Environment:**

The cloud software infrastructure layer provides an abstraction level for basic IT resources that are offered as services to higher layers: computational resources (usually VMs), storage, and (network) communication. These services can be used individually, as is typically the case with storage services, but they’re often bundled such that servers are delivered with certain network connectivity and (often) access to storage. This bundle, with or without storage, is usually referred to as IaaS.

**Computational Resources:**

A highly relevant set of computational resource vulnerabilities concerns how virtual machine images are handled: the only feasible way of providing nearly identical server images—thus providing on-demand service for virtual servers—is by cloning template images.

Because cryptography is frequently used to overcome storage-related vulnerabilities, this core technology’s vulnerabilities—insecure or obsolete cryptography and poor key management—play a special role for cloud storage.
Communication:

The most prominent example of a cloud communications service is the networking provided for VMEs in an IaaS environment. Because of resource pooling, several customers are likely to share certain network infrastructure components: vulnerabilities of shared network infrastructure components, such as vulnerabilities in a DNS server, Dynamic Host Configuration Protocol, and IP protocol vulnerabilities, might enable network-based cross-tenant attacks in an IaaS infrastructure.

Cloud Web Applications:

A Web application uses browser technology as the front end for user interaction. With the increased uptake of browser-based computing technologies such as JavaScript, Java, Flash, and Silverlight, a Web cloud application falls into two parts:

• An application component operated somewhere in the cloud, and
• A browser component running within the user’s browser.

Identity, Authentication, Authorization, and Auditing Mechanisms:

Most vulnerability associated with the IAAA component must be regarded as cloud-specific because they’re prevalent in state-of-the-art cloud offerings. Earlier, we gave the example of weak user authentication mechanisms; other examples include

• Denial of service by account lockout:
  
  One often-used security control—especially for authentication with username and password—is to lock out accounts that have received several unsuccessful authentication attempts in quick succession. Attackers can use such attempts to launch DoS attacks against a user.

• Weak credential-reset mechanisms:

  When cloud computing providers manage user credentials themselves rather than using federated authentication, they must provide a mechanism for resetting credentials in the case of forgotten or lost credentials. In the past, password-recovery mechanisms have proven particularly weak.

• Insufficient or faulty authorization checks:

  State-of-the-art Web application and service cloud offerings are often vulnerable to insufficient or faulty authorization checks that can make unauthorized information or actions available to users. Missing authorization checks, for example, are the root cause of URL-guessing attacks. In such attacks, users modify URLs to display information of other user accounts.

• Coarse authorization control:

  Cloud services’ management interfaces are particularly prone to offering authorization control models that are too coarse. Thus, standard security measures, such as duty separation, can’t be implemented because it’s impossible to provide users with only those privileges they strictly require to carry out their work.

• Insufficient logging and monitoring possibilities:

  Currently, no standards or mechanisms exist to give cloud customers logging and monitoring facilities within cloud resources. This gives rise to an acute problem: log files record all tenant events and can’t easily be pruned for a single tenant. Also, the provider’s security monitoring is often hampered by insufficient monitoring capabilities. Until we develop and implement usable logging and monitoring standards and facilities, it’s difficult—if not impossible—to implement security controls that require logging and monitoring.

  Of all these IAAA vulnerabilities, in the experience of cloud service providers, currently, authentication issues are the primary vulnerability that puts user data in cloud services at risk.

Provider:

Vulnerabilities that are relevant for all cloud computing components typically concern the provider—or rather users inability to control cloud infrastructure as they do their own infrastructure. Among the control challenges are insufficient security audit possibilities, and the fact that certification schemes and security metrics aren’t adopted to cloud computing. Further, standard security controls regarding audit, certification, and continuous security monitoring can’t be implemented effectively.

CLOUD COMPUTING TECHNOLOGIES:

Cloud computing builds heavily on capabilities available through several core technologies:

• Web applications and services

  Software as a service (SaaS) and platform as a service (PaaS) are unthinkable without Web application and Web services technologies: SaaS offerings are typically implemented as Web applications, while PaaS offerings provide development and runtime environments for Web applications and services. For infrastructure as a service (IaaS) offerings, administrators typically implement associated services and APIs, such as the management access for customers, using Web application/service technologies.
• **Virtualization IaaS offerings**

These technologies have virtualization techniques at their very heart; because PaaS and SaaS services are usually built on top of a supporting IaaS infrastructure, the importance of virtualization also extends to these service models. In the future, we expect virtualization to develop from virtualized servers toward computational resources that can be used more readily for executing SaaS services.

• **Cryptography**

Many cloud computing security requirements are solvable only by using cryptographic techniques.

As cloud computing develops, the list of core technologies is likely to expand

**ESSENTIAL CHARACTERISTICS:**

In its description of essential cloud characteristics, the US National Institute of Standards and Technology (NIST) captures well what it means to provide IT services from the conveyor belt using economies of scale:

• **On-demand self-service**

Users can order and manage services without human interaction with the service provider, using, for example, a Web portal and management interface. Provisioning and de-provisioning of services and associated resources occur automatically at the provider.

• **Ubiquitous network access**

Cloud services are accessed via the network (usually the Internet), using standard mechanisms and protocols.

• **Resource pooling**

Computing resources used to provide the cloud service are realized using a homogeneous infrastructure that’s shared between all service users.

• **Rapid elasticity** - Resources can be scaled up and down rapidly and elastically.

• **Measured service** - Resource/service usage is constantly metered, supporting optimization of resource usage, usage reporting to the customer, and pay-as-you-go business models.

**CORE-TECHNOLOGY VULNERABILITIES:**

Cloud computing’s core technologies—Web applications and services, virtualization, and cryptography—have vulnerabilities that are either intrinsic to the technology or prevalent in the technology’s state-of-the-art implementations. Three examples of such vulnerabilities are virtual machine escape, session riding and hijacking, and insecure or obsolete cryptography.

First, the possibility that an attacker might successfully escape from a virtualized environment lies in virtualization’s very nature. Hence, we must consider this vulnerability as intrinsic to virtualization and highly relevant to cloud computing.

Second, Web application technologies must overcome the problem that, by design, the HTTP protocol is a stateless protocol, whereas Web applications require some notion of session state. Many techniques implement session handling and—as any security professional knowledgeable in Web application security will testify—many session handling implementations are vulnerable to session riding and session hijacking. Whether session riding/hijacking vulnerabilities are intrinsic to Web application technologies or are “only” prevalent in many current implementations is arguable; in any case, such vulnerabilities are certainly relevant for cloud computing.

Finally, crypto analysis advances can render any cryptographic mechanism or algorithm insecure as novel methods of breaking them are discovered. It’s even more common to find crucial flaws in cryptographic algorithm implementations, which can turn strong encryption into weak encryption (or sometimes no encryption at all). Because broad uptake of cloud computing is unthinkable without the use of cryptography to protect data confidentiality and integrity in the cloud, insecure or obsolete cryptography vulnerabilities are highly relevant for cloud computing.

**ESSENTIAL CLOUD CHARACTERISTIC VULNERABILITIES:**

As we noted earlier, NIST describes five essential cloud characteristics: on-demand self-service, ubiquitous network access, resource pooling, rapid elasticity, and measured service.

Following are examples of vulnerabilities with root causes in one or more of these characteristics:

• **Unauthorized access to management interface**

The cloud characteristic on-demand self-service requires a management interface that’s accessible to cloud service users. Unauthorized access to the management interface is therefore an especially relevant vulnerability for cloud systems: the probability that unauthorized access could occur is much higher than for traditional systems where the management functionality is accessible only to a few administrators.
Cloud Specific Issues and Vulnerabilities Solutions

- **Internet protocol vulnerabilities**

  The cloud characteristic ubiquitous network access means that cloud services are accessed via network using standard protocols. In most cases, this network is the Internet, which must be considered untrusted. Internet protocol vulnerabilities—such as vulnerabilities that allow man-in-the-middle attacks—are therefore relevant for cloud computing.

- **Data recovery vulnerability**

  The cloud characteristics of pooling and elasticity entail that resources allocated to one user will be reallocated to a different user at a later time. For memory or storage resources, it might therefore be possible to recover data written by a previous user.

- **Metering and billing evasion**

  The cloud characteristic of measured service means that any cloud service has a metering capability at an abstraction level appropriate to the service type (such as storage, processing, and active user accounts). Metering data is used to optimize service delivery as well as billing. Relevant vulnerabilities include metering and billing data manipulation and billing evasion.

  Thus, we can leverage NIST’s well-founded definition of cloud computing in reasoning about cloud computing issues.

**Defects in Known Security Controls**

Vulnerabilities in standard security controls must be considered cloud specific if cloud innovations directly cause the difficulties in implementing the controls. Such vulnerabilities are also known as control challenges.

Here, we treat three examples of such control challenges. First, virtualized networks offer insufficient network-based controls. Given the nature of cloud services, the administrative access to IaaS network infrastructure and the ability to tailor network infrastructure are typically limited; hence, standard controls such as IP-based network zoning can’t be applied. Also, standard techniques such as network-based vulnerability scanning are usually forbidden by IaaS providers because, for example, friendly scans can’t be distinguished from attacker activity. Finally, technologies such as virtualization mean that network traffic occurs on both real and virtual networks, such as when two virtual machine environments (VMEs) hosted on the same server communicate. Such issues constitute a control challenge because tried and tested network-level security controls might not work in a given cloud environment.

The second challenge is in poor key management procedures. As noted in a recent European Network and Information Security Agency study,3 cloud computing infrastructures require management and storage of many different kinds of keys. Because virtual machines don’t have a fixed hardware infrastructure and cloud-based content is often geographically distributed, it’s more difficult to apply standard controls—such as hardware security module (HSM) storage—to keys on cloud infrastructures.

Finally, security metrics aren’t adapted to cloud infrastructures. Currently, there are no standardized cloud-specific security metrics that cloud customers can use to monitor the security status of their cloud resources. Until such standard security metrics are developed and implemented, controls for security assessment, audit, and accountability are more difficult and costly, and might even be impossible to employ.

**PREVALENT VULNERABILITIES IN STATE-OF-THE-ART CLOUD OFFERINGS:**

Injection vulnerabilities are exploited by manipulating service or application inputs to interpret and execute parts of them against the programmer’s intentions. Examples of injection vulnerabilities include

- SQL injection, in which the input contains SQL code that’s erroneously executed in the database back end;
- Command injection, in which the input contains commands that are erroneously executed via the OS; and
- Cross-site scripting, in which the input contains JavaScript code that’s erroneously executed by a victim’s browser.

In addition, many widely used authentication mechanisms are weak. For example, usernames and passwords for authentication are weak due to

- Insecure user behavior (choosing weak passwords, reusing passwords, and so on), and
- Inherent limitations of one-factor authentication mechanisms.

Also, the authentication mechanisms’ implementation might have weaknesses and allow, for example, credential interception and replay. The majority of Web applications in current state-of-the-art cloud services employ usernames and passwords as authentication mechanism.

**SECURITY ISSUES AND SOLUTIONS IN CLOUD COMPUTING:**

This paper concerns security issues and solutions in cloud computing. Cloud computing is a catch-all phrase that covers virtualized operating systems running on virtual hardware on untold numbers of physical servers.
The cloud term has consumed High-Performance Computing (HPC), Grid computing and Utility Computing. The Cloud Security Alliance has adopted the definition developed by NIST; a computing in the cloud is a model exhibiting the following characteristics, on-demand self-service, Broad Network Access, Resource pooling, and Rapid elasticity and Measured service (Cloud Security Alliance Guidance Version 2.1, 2009, p. 15). This is an area that appears to be growing larger and more pervasive as the benefits of cloud architectures become better understood. More organizations start their own cloud projects and more application developers sign on for cloud development as the hyperbole is shaken out and the real parameters of the key technologies are discovered and perfected. The basic areas of cloud vulnerability are similar to the standard issues that surround networking and networked applications. The issues specific to cloud architectures include network control being in in the hands of third parties and a potential for sensitive data to be available to a much larger selection of third-parties, both on the staff of the cloud providers, and among the other clients of the cloud.

The quick adoption of the cloud model is plain in the success of the Amazon Elastic Cloud Computing (EC2) product, the buy-in from IBM with their backing of the highly concurrent, massively parallel language X-10 (Saraswat, Vijay, 2010) and Microsoft’s investment in its Azure cloud (Quet al., 2009). Janine Milne reported that eight of ten businesses surveyed in the UK were opting for private cloud initiatives rather than public cloud projects and they stated the issues of concern to be data security in transit, in storage or during processes (Milne, 2010). It is plain that the field is full and the harvest for the IT security profession and IT in general are excellent.

The literature available on cloud security is plentiful, and there is enough higher-quality work to develop a conceptual framework for security issues and solutions.

Security Solutions:

There are several groups interested in developing standards and security for clouds and cloud security. The Cloud Security Alliance (CSA) is gathering solution providers, non-profits and individuals to enter into discussion about the current and future best practices for information assurance in the cloud (Cloud Security Alliance (CSA) – security best practices for cloud computing, 2009). The Cloud Standards web site is collecting and coordinating information about cloud-related standards under development by other groups (Clouds Standards, 2010). The Open Web Application Security Project (OWASP) maintains a top 10 list of vulnerabilities to cloud-based or Software as a Service deployment models which is updated as the threat landscape changes (OWASP, 2010). The Open Grid Forum publishes documents to containing security and infrastructural specifications and information for grid computing developers and researchers (Open Grid Forum, 2010).

Web Application Solutions

The best security solution for web applications is to develop a development framework that shows and teaches a respect for security. Tsai, W., Jin, Z., & Bai, X. (2009) put forth a four-tier framework for web-based development that though interesting, only implies a security facet in the process (Tsai, Jin, & Bai, 2009, p. 1). Towards best practices in designing for the cloud by Berre, Roman, Landre, Heuvel, SkÅ, Udn, Lennon, & Zeid (2009) is a road map toward cloud-centric development (Berre et al., 2009), and the X10 language is one way to achieve better use of the cloud capabilities of massive parallel processing and concurrency (Saraswat, Vijay, 2010).

Accessibility Solutions

KrÄngel, C., Toth, T., & Kirda, E. (2002) point out the value of filtering a packet-sniffer output to specific services as an effective way to address security issues shown by anomalous packets directed to specific ports or services (KrÄngel et al., 2002).

(KrÄngel et al., 2002) An often-ignored solution to accessibility vulnerabilities is to shut down unused services, keep patches updated, and reduce permissions and access rights of applications and users.

Authentication Solutions

Halton and Basta (2007) suggest one way to avoid IP spoofing by using encrypted protocols wherever possible. They also suggest avoiding ARP poisoning by requiring root access to change ARP tables; using static, rather than dynamic ARP tables; or at least make sure changes to the ARP tables are logged. (Basta & Halton, 2007, p. 166).

Data Verification, Tampering, Loss and Theft Solutions

Raj, Nathuji, Singh and England (2009) suggest resource isolation to ensure security of data during processing, by isolating the processor caches in virtual machines, and isolating those virtual caches from the Hypervisor cache (Raj, Nathuji, Singh, & England, 2009, p. 80). Hayes points out that there is no way to know if the cloud providers properly deleted a client’s purged data, or whether they saved it for some unknown reason (Hayes, 2008, p. 11). Would cloud-providers and clients have custody battles over client data?
Privacy and Control Solutions

Hayes (2008) points out an interesting wrinkle here, allowing a third-party service to take custody of personal documents raises awkward questions about control and ownership: If you move to a competing service provider, can you take a data with you? Could you lose access to a document if you fail to pay a bill? (Hayes, 2008, p. 11). The issues of privacy and control cannot be solved, but merely assured with tight service-level agreements (SLAs) or by keeping the cloud itself private.

Physical access solutions

One simple solution, which Milne (2010) states to be a widely used solution for UK businesses is to simply use in-house private clouds (Milne, 2010). Nurmi, Wolski, Grzegorczyk, Obertelli, Soman, Youseff, & Zagorodnov show a preview of one of the available home-grown clouds in their (2009) presentation. The Eucalyptus Open-Source Cloud-Computing System (Nurmi et al., 2009).

CONCLUSION:

Cloud computing is in constant development; as the field matures, additional cloud-specific vulnerabilities certainly will emerge, while others will become less of an issue. Using a precise definition of what constitutes a vulnerability from the Open Group’s risk taxonomy and the four indicators of cloud-specific vulnerabilities we identify here offers a precision and clarity level often lacking in current discourse about cloud computing security. Control challenges typically highlight situations in which otherwise successful security controls are ineffective in a cloud setting. Thus, these challenges are of special interest for further cloud computing security research. Indeed, many current efforts—such as the development of security metrics and certification schemes, and the move toward full-featured virtualized network components—directly address control challenges by enabling the use of such tried-and-tested controls for cloud computing.

ACKNOWLEDGEMENTS:

The authors would like to acknowledge S.K.Prasanth, SK.Lokesh for the useful discussions on this topic and their inputs and feedback while writing this paper.

REFERENCES:


A New and Efficient Implementation of Encryption Based on Asymmetric and Symmetric Ciphers

K. Sasi Priyadarshini & T. Sanath Kumar
ECE Department, ASCET, India
E-mail: Sasi.priyadarshini@gmail.com, sanath.ece@audisankara.com

Abstract – Bluetooth technology is an emerging wireless networking standard, which is based on chip that provides short-range wireless frequency hopping communication. It is mainly applied to the communication between mobile terminal devices, such as palm computers, mobile phones, laptops and so on, and also can successfully simplify the communication among above devices and the Internet, so that the data transmission between these modern communication equipments and Internet has become more quickly and efficiently, and widen the road for wireless communications. For the security of data transmission in Bluetooth communication, a hybrid encryption algorithm based on DES and RSA is proposed. In the proposed encryption algorithm, instead of the E0 encryption, DES algorithm is used for data transmission because of its higher efficiency and RSA algorithm is used for the encryption of the key of the DES because of its advantages in key cipher. Under the dual protection with the DES and the RSA algorithm, the data transmission in the Bluetooth system will be more secure.

Keywords - Bluetooth; E0 key stream; hybrid encryption algorithm; data transmission.

I. INTRODUCTION

Bluetooth technology has the characteristic of wireless, openness, low power and so on. However, the phenomenon of data-leaking frequently arise in using the Bluetooth technology for data transfer, since the emergence of Bluetooth, even if the Bluetooth takes the very robust security measures, there are still serious security risks. The encryption algorithm using in Bluetooth encryption process is the E0 stream cipher. However, this algorithm has some shortcomings, 128-bit E0 stream ciphers in some cases can be cracked by 0 (264) mode in some cases. So, for most applications that which need to give top priority to confidentiality, the data security is not enough if only use Bluetooth. Now I will introduce the Bluetooth mechanism, its disadvantages, and then propose a hybrid encryption algorithm to solve the current security risk in Bluetooth data transmission.

II. THE ENCRYPTION ALGORITHM IN BLUETOOTH SECURITY MECHANISM

A. Bluetooth security mechanism

The Bluetooth specification defines three security modes:

1) Safe Mode 1: No safe mode, which has the lowest security level;
2) Safe Mode 2: service-oriented security model, which start after the establishment of the channel;
3) Safe Mode 3: link-oriented security model, which install and initial before communication link is established.

Bluetooth system provides safety precautions in the application layer and link layer, the two sides achieve authentication and encryption in the same way. Link layer uses four entities to ensure the safety:

1) 48-bit of the Bluetooth device address, which is global uniqueness decided by the IEEE;
2) The authentication key for entity authentication is 128-bit;
3) The secret key for data encryption is 8 ~ 128-bit;
4) 128-bit random number trades once, changes once.

Two keys are generated in the initialization process and do not open, encryption key is generated in the certification process from the authentication key, but it is different from the authentication key, every time when you activate the encryption, it will generate a new secret key. Authentication key is more stable, after generating, it is decided by the concrete application of
Bluetooth device whether to change Now, Applying linear congruential generator to generate random numbers is widely adopted at present, its expression is as follows:

\[ X_{n+1} = a \times X_n + c \pmod{k}, \quad n \geq 0, \quad (1) \]

Where \( a, c \) are constants, \( k \) is the mold, we generate a series of random numbers taking a certain number of \( X_0 \) for seed number.

B. Authentication and encryption process of Bluetooth

Bluetooth security mechanism is divided into three modules including key generation, authentication and encryption, and adopt four kinds of algorithms as E0, E1, E2, E3. Bluetooth system provides authentication, encryption and key management functions in Link layer. PIN code was entered by the user, by means of the E2 algorithm for generating the link key, by means of E3 algorithm, getting encryption key, make use of E0 algorithm generated key stream, and encrypt plaintext, then get cipher text. Figure 1 is the process of Bluetooth encryption. Figure 1. the process of Bluetooth encryption.

![Fig. 1 : The process of Bluetooth encryption.](image)

The three modules of Figure 1 are as follows: 1) key generation module, algorithm E2 is used for generating the link key, and its input parameter is a 4-digit passwords number which is entered by the user, the algorithm E3 calculates encryption key KC by the use of E2 link key encryption key as input parameters. 2) Encryption module, algorithm E0 can be used for generating keys stream to encrypt the original data. 3) Authentication module, algorithm E1 is the crucial algorithms in the authentication process, the two units in need of certification use each authentication algorithm E1 to generate identification word and compare, then complete certification.

A. Analysis of E0 Algorithms

E0 algorithm is the encryption algorithms in Bluetooth link layer, which belongs to stream encryption method, that is to say it take data flow and the key bit stream Exclusive-or operation. The payload of each packet is encrypted separately, and the encryption occurs before MPE-FEC, after the cyclic redundancy check. The main principle is to use linear feedback shift register to generate pseudo-random sequence, after that form key stream that can be used for encryption, and then take the key stream and data stream that need encryption Exclusive-or operation, and achieve encryption. During decryption, the cipher text take Exclusive-or operation once more, re-plaintext can be obtained.

III. HIDDEN DANGER OF BLUETOOTH SECURITY SYSTEM

A. The weakness of E0 stream cipher algorithm the main weakness of Stream cipher algorithm is that if a pseudo-random sequence make an error, it will make the whole cipher text mistake happen, it also bring about the cipher text can not restore back to plaintext in decipherment. If its output is endless sequence of 0, then the cipher text is the plaintext, so that the whole system is worthless; if its output is a periodic 16-bit mode, then the algorithm is only an Exclusive-or operation which can ignore security; if the output is a series of endless random sequence (which is truly random, non-pseudo-random), then there is one-time pad and very perfect safety Secret keys that output from key stream generator output is the closer to random, it is much harder for the cryptanalyst. However, this random key stream is not acquired easily.

B. Limited resources capacity of linear feedback shift register LFSR

Encryption algorithm used in Bluetooth technology standard is somewhat fragile, and even if its E0 stream cipher uses 128-bit key, in some cases, the complexity of their decoding is only 0. There are 4 LFSR in key stream generator of E0 stream cipher. If a certain LFSR of the key stream generator generated a sequence of cycle is shorter than the key, there is the threat from attacker divide and conquer. In fact, the software implementation of LFSR algorithm is not faster than the DES and RSA hybrid encryption algorithm.

a. Low credibility of PIN

Bluetooth technology uses non-standard 4-digit PIN code and another variable to generate the link key and encryption key. Actually, 4-digit PIN code is the only variable which is the real key generated, resulting only one key (a random number) transport in the air. Here brute force attack is exhaustive key search, if the PIN is k bits, then only in the case of cipher text attack, an attacker can search the value of the PIN through 2^k
times. Therefore, the credibility of the PIN code is lower, 4 bits PIN code only has 10,000 possibilities. One solution is to choose and use 16-byte PIN code, or use the public key system. If using a longer PIN code can increase the difficulty of the attacker get encryption keys, but it is really very inconvenient, because every time when a secure connection is established, we should have to enter a PIN code.

b. High probability of non-link key cheat

Along with the use of the link key takes new problems. Authentication and encryption set up on the basis of the link key. All the other Information used in this connection usually is public. However, this will lead to the following questions:

1) Device A and B using the secret key of device A as the link key.
2) At the same time or later, device C may communicate with the device A and use the key of device A as their link key.
3) Device B can use the link key of device A to decrypt the communication information between device A and device C.

As discussed above, device B which get the key of device A can use this key with one camouflaged BD-ADDR to calculate the encryption key, then achieve monitoring device A to communicate with other devices. And the device B can disguise device C through device A certifies, also can disguise device A through device C certifies.

c. Address Spoofing

Every Bluetooth device has a unique Bluetooth device address. However, its uniqueness raises new problems. Once the ID links with a certain fixed person, this person can be tracked and their activities can easily be recorded. In this case, the individual's privacy will be violated. Above these problems can lead people to believe that the Bluetooth security system is highly unreliable, but there is a fact can not be ignored is that: in general, the data transmitted via Bluetooth connection is not very important. Now, Bluetooth standard is only applicable in smaller networks because considered security technology, Bluetooth technology provides data security measures for small-scale applications appear to be enough, but any sensitive data or the data that may cause problems should not be transferred via Bluetooth directly.

IV. THE IDEAS AND PROCESSES OF HYBRID ENCRYPTION ALGORITHM

RSA algorithm is the first relatively complete public key algorithm. It can be used for data encryption, also can be used for digital signature algorithms. RSA cryptosystem is based on the difficulty of integer factorization in the group Zn, and its security establishes in the assumption that constructed by almost all the important mathematicians, it is still a theorem that does not permit, which is lack of proof, but Mathematicians believe it is existent.

DES is a group cipher algorithm, which encrypts data by a group of 64-bit. A group of 64-bit plaintext is entered from one beginning of the algorithm. 64-bit cipher text is exported from the other side. DES is a symmetric algorithm, encryption and decryption use the same algorithm (e the different key arrangement), the key can be any 56-bit value (the key is usually 64-bit binary number, but every number that is a multiple of 8-bit used for parity are ignored). This algorithm uses two basic encryption techniques, make them chaos and spread, and composite them. Seeing from the efficiency of encryption and decryption,

DES algorithm is better than the RSA algorithm. The speeds of DES encryption is up to several M per second, it is suitable for encrypting large number of message; RSA algorithm is based on the difficulty of factoring, and its computing velocity is slower than DES', and it is only suitable for encrypting a small amount of data, The RSA encryption algorithm used in the .NET, it encrypts data at most 117 bytes of once. Seeing from key management, RSA algorithm is more superior than the DES algorithm. Because the RSA algorithm can distribute encryption key openly, it is also very easy to update the encryption keys, and for the different communication objects, just keep the decryption keys secret; DES algorithm requires to distribute a secret key before communication, replacement of key is more difficulty, different communication objects, DES need to generate and keep a different key. Based on the comparison of above DES algorithm and RSA algorithms, in order to give expression to the advantages of the two algorithms, and avoid their shortcomings at the same time, we can conceive a new encryption algorithm, that is, DES and RSA hybrid encryption algorithm. We will apply hybrid encryption algorithm to Bluetooth technology, we can solve the current security risks of Bluetooth technology effectively. The entire hybrid encryption process is as follows: Let the sender is A, the receiver is B, B's public key is eB, B's private key is dB, K is DES encryption session key (assuming that the two sides of communication know each RSA public key).

A. Process of encryption

During the process of sending encrypted information, the random number generator uses 64-bit DES session key only once, it encrypt the plaintext to produce cipher text. On the other hand, the sender get
A New and Efficient Implementation of Encryption Based on Asymmetric and Symmetric Ciphers

debite’s public key from public key management center, and then using RSA to encrypt session key. Finally, the combination of the session key from RSA encryption and the cipher text from DES encryption are sent out. Key stream has the characteristics of self-synchronization, if the key text which is sent encounter errors and data lost, it will only affect a small section of the final text (64-bit). The first, DES algorithm encrypts Bluetooth data packet:

1) Bluetooth packet plaintext M is divided into 64-bit plaintext Mi (i=1,2,…,n)).
2) Encrypts Mi for 16 cycles by 64-bit key K, and Mi will turn into a 64-bit cipher text Ci (i = 1, 2, ..., n), then all the Ci (i = 1, 2, ..., n) are combined into cipher text C. The second, RSA algorithm encrypts the key of DES algorithm:
3) Obtain RSA public key of receiver B from the key server, or other sources
4) Make DES 64-bit session key K for RSA encryption by public key eB that obtains from recipient, then a session key encrypted information CK is formed.
5) Composite Cipher text message C from the use of DES encryption, and session key CK from RSA encryption, we can get the hybrid CM for transmission. Figure 2 is the whole mixed-encryption process.

Fig. 2: The whole mixed encryption process

B. Process of decryption

The decryption of hybrid encryption algorithm is as follows. The first, the receiver B divide received cipher text CM into two parts, one is cipher text CK from the RSA algorithm encryption, the other is cipher text C from the DES algorithm encryption. The second, the receiver B decrypt cipher text CK by their own private key dB, receive the key K which belongs DES algorithm, then decrypt the cipher text C to the original M by key K. Figure 3 is a decryption of hybrid encryption algorithm.

Fig. 3: A decryption of hybrid encryption algorithm

C. The advantages of hybrid encryption algorithm

- Using RSA algorithm and the DES key for data transmission, so it is no need to transfer DES key secretly before communication;
- Management of RSA key is the same as RSA situation, only keep one decryption key secret;
- Using RSA to send keys, so it can also use for digital signature;
- The speed of encryption and decryption is the same as DES. In other words, the time-consuming RSA just do with DES keys;

D. Analysis Of Hybrid Encryption Algorithm

Safety of Hybrid encryption algorithm is based on the safety of RSA algorithm and DES algorithm, operating efficiency of hybrid encryption algorithm depends on the speed and high efficiency of encryption and decryption by DES algorithm. Of course, the Bluetooth based on hybrid encryption algorithm, its data transmission security depend on the security of hybrid encryption algorithm. At present, RSA encryption algorithm is a kind of more successful public key cryptosystem in theoretical and practical application, and its security is based on the difficulty of large integer resolution into prime factors. And its security depends on the large integer factorization, but whether it is equivalent to large integer factorization has not been proven in theory, because there is no proof of cracking RSA will definitely need to make large integer factorization. Because of the dual protection of DES algorithm and RSA algorithm, the data in transit is safe.

V. CONCLUSIONS

Bluetooth technology is a new technology, which will our transmission method. However, the Bluetooth technology has not fully considerate security issues in the standardization process. As communication networks, it uses wireless channel for the transmission medium. Compared to the fixed network Bluetooth network is more vulnerable to be attacked. For the applications that take data security as priori, achieving a high level of data security is essential. Currently, stream
cipher E0 used in Bluetooth standard has many shortcomings, while the DES and RSA hybrid encryption algorithm is relatively more secure and easier to achieve, thus ensures data transmission between the Bluetooth device safety and real-time.

ACKNOWLEDGMENT
The authors thank editor-in-chief and anonymous reviewers for their comments and suggestions. The work is supported by the Science & Technology Research Program of Zhejiang Province, China (No.2009C03016-4, 2008C01060-2).

REFERENCES
Combination of Density Based and Partition Based Clustering Algorithm-DBKmeans

B.L.krishna, P.Jhansi Lakshmi & P.Satya Prakash
Department of computer science, Vignan University, Vadlamudi, Guntur, India
E-mail: laxmanbodduluri@gmail.com, laxmi.jhansi@gmail.com Satya.softguy@gmail.com

Abstract – Mining or extracting the knowledge from the large amount of data is known as data mining. Here, the collection of data increases exponentially so that for extracting the efficient data we need good methods in data mining. Data mining analyzes several methods for extracting the data. Clustering is one of the methods for extracting the data from large amount of data. Multiple clustering algorithms were developed for clustering. Multiple clustering can be combined so that the final partitioning of data provides better clustering. Efficient density based k-means clustering algorithm has been proposed to overcome the drawbacks of dbscan and k-means clustering algorithms. The algorithm performs better than dbscan while handling clusters of circularly distributed data points and slightly overlapped clustering.

Keywords - clustering analysis, k-means, and dbscan.

I. INTRODUCTION
Cluster analysis divides data into meaningful or useful groups (clusters). If meaningful clusters are the goal, then the resulting clusters should capture the “natural” structure of the data. For example, cluster analysis has been used to group related documents for browsing, to find genes and proteins that have similar functionality, and to provide a grouping of spatial locations prone to earthquakes. However, in other cases, cluster analysis is only a useful starting point for other purposes, e.g., data compression or efficiently finding the nearest neighbors of points. Whether for understanding or utility, cluster analysis has long been used in a wide variety of fields: psychology and other social sciences, biology, statistics, pattern recognition, information retrieval, machine learning, and data mining.

What Cluster Analysis Is
Cluster analysis groups objects (observations, events) based on the information found in the data describing the objects or their relationships. The goal is that the objects in a group will be similar (or related) to one other and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group, and the greater the difference between groups, the “better” or more distinct the clustering.

DISTANCE MEASURES
The number of points in a dataset is denoted by N. Each point is denoted by Pi, Pj and so on. k denotes the number of clusters and d denotes the number of dimensions of a point. D denotes the set of dimensions and Dix, Djy represent the subsets of dimensions of the points Pi and Pj respectively. l,m and x are simply used as indices.

EUCLIDEAN DISTANCE
An N x N matrix Me is calculated. For points with d dimensions, the Euclidean distance Me(pi, pj) between two points Pi and Pj is defined as follows:

$$M_e (p_i, p_j) = \sqrt{\sum_{z=1}^{d} (p_{iz} - p_{jz})^2}$$
where $P_i$ and $P_j$ represent the xth dimension values of $P_i$ and $P_j$ respectively. Also, $M_e$ is a symmetric matrix.

II. RELATED WORK

K-MEANS:

The term "k-means" was first used by James Mac Queen in 1967, though the idea goes back to Hugo Steinhaus in 1956. The standard algorithm was first proposed by Stuart Lloyd in 1957 as a technique for pulse-code modulation, though it wasn’t published until 1982.

In statistics and machine learning, k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data as well as in the iterative refinement approach employed by both algorithms.

The kmeans implementation is easy when compared to the other clustering algorithms. This simply involves the selection of the initial indices and the calculation of the distances to these indices to every other point in the data base and there by forming the clusters. This is a continuous procedure and the procedure is repeated until the same means for all the clusters are repeated.

KMEANS ALGORITHM:

```cpp
void cluster( Data D, clno k, Num n )
{
  If( k <=n)
  {
    Randomly pick k-objects as initial clusters
    Repeat:
    {
      For all n objects calculate the distance
      and assign the clusters
      For obtained clusters calculate Means M_i &
      update clusters C_i
    }
    Until: no change in Means ( i=1 to k M_i==M_i-1 )
  }
  Else
    Clustering not possible
}
```

Where

- $K$ - No of clusters.
- $D$ - Data set taken.
- $N$ - Size of the data set (N by 2)
- $M_i$ - Means to the clusters.
- $C_i$ - Clusters formed.

From the algorithm it’s clear that if that if the no of clusters required is greater than the size of the data it’s not possible to cluster the data. The actual implementation is seen in the next section.

KMEANS DATA FLOW DIAGRAM

K MEANS ADVANTAGES

- With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small).
- K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular.

KMEANS DISADVANTAGES

- Difficulty in comparing quality of the clusters produced (e.g. for different initial partitions or values of K affect outcome).
- Fixed number of clusters can make it difficult to predict what K should be.
- Does not work well with non-globular clusters.
- Different initial partitions can result in different final clusters. It is helpful to rerun the program using the same as well as different K values, to compare the results achieved.
PROBLEMS WITH K-MEANS

- When the numbers of data are not so many, initial grouping will determine the cluster significantly.
- The result is circular cluster shape because based on distance.
- The number of cluster, K, must be determined beforehand. Selection of value of K is itself an issue and sometimes its hard to predict beforehand the number of clusters that would be there in data.
- We never know the real cluster, using the same data, if it is inputted in a different order may produce different cluster if the number of data is few.
- Sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the local optimum.
- We never know which attribute contributes more to the grouping process since we assume that each attribute has the same weight.

DBSCAN:

DBScan (Density-Based Spatial Clustering of Applications with Noise) is a data clustering algorithm proposed by Martin Ester, Hans Peter Kriegel, Jorg Sander and Xiaowei in 1996. It is a density-based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes. DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. The main implementation of this algorithm depends up on the selection of the nearest neighbors. This is done based on the given density & the distance value. This is the finest algorithm used to find the outliers.

DENSITY REACHABILITY AND DENSITY CONNECTIVITY:

Density reachability is the first building block in dbscan. It defines whether two distance close points belong to the same cluster. Points p1 is density reachable from p2 if two conditions are satisfied: (i) the points are close enough to each other: distance (p1, p2) <=e, (ii) there are enough of points in is neighborhood: |{ r : distance(r,p2)}|>=m, where r is a database point. Density connectivity is the last building step of dbscan. Points p0 and pn are density connected, if there is a sequence of density reachable points p1,i2,...i(n-1) from p0 to pn such that p(i+1) is density reachable from pi. A dbscan cluster is a set of all density connected points.

EXPLANATION OF DBSCAN STEPS:

- DBScan requires two parameters: epsilon (eps) and minimum points (minPts). It starts with an arbitrary starting point that has not been visited. It then finds all the neighbor points within distance eps of the starting point.
- If the number of neighbors is greater than or equal to minPts, a cluster is formed. The starting point and its Neighbors are added to this cluster and the starting point is marked as visited. The algorithm then repeats the Evaluation process for all the neighbors recursively.
- If the number of neighbors is less than minPts, the point is marked as noise.
- If a cluster is fully expanded (all points within reach are visited) then the algorithm proceeds to iterate through the remaining unvisited points in the dataset.

DBSCAN ALGORITHM:

```c
void dbscan(Data D, minpts min, eps ep)
{
    Repeat:
    If i==1 select as default index else Select randomly d_i as initial index
    Repeat:
    Get neighbours (d_i, min, ep):
    For each neighbour: Mark as visited for that loop
    Get neighbours (d_i, min, ep):
    (UNTIL) : no neighbour is obtained;
    if( minpoints present )
    add to cluster else leave & mark as unvisited;
    Until all the nodes are visited;
}
```

Where D - Data set.
Min-Minimum points required to form a cluster.
Ep - Minimum distance required to select as a neighbor.

DBSCAN ADVANTAGES

1. DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k-means.
2. DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect
(different clusters being connected by a thin line of points) is reduced.

3. **DBSCAN has a notion of noise.**

4. **DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database.** (However, points sitting on the edge of two different clusters might swap cluster membership if the ordering of the points is changed, and the cluster assignment is unique only up to isomorphism.)

**DBSCAN DISADVANTAGES**

- The quality of DBSCAN depends on the distance measure used in the function `regionQuery(P,ε)`. The most common distance metric used is Euclidean distance. Especially for high-dimensional data, this metric can be rendered almost useless due to the so-called "Curse of dimensionality", making it difficult to find an appropriate value for ε. This effect, however, is also present in any other algorithm based on Euclidean distance.

- DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-ε combination cannot then be chosen appropriately for all clusters.

**III. PROBLEM IDENTIFICATION**

Some type clusters are formed in below figure format by using dbscan.

![Circularly over lapped clusters](image)

Fig. 2: circularly over lapped clusters

In order to avoid this type of circularly overlapped clusters in dbscan we have to follow the dbkmeans

**IV. IMPLEMENTATION OF PROPOSED WORK DBKMEANS**

The DBKMEANS combines the both kmeans and dbscan algorithms. In order to overcome the drawbacks of the both algorithms.

```c
void DBKMEANS(c1no k , min ep , dist ε, Dataset D)
{
    c=0
    for each unvisited point p in dataset D
    {
        N = getNeighbors (p, ε)
        if (sizeof(N) < ep)
            mark p as NOISE
        else
            ++ c
            mark p as visited
            add p to cluster c
        recurse (N)
    }
    Now will have m clusters
    for each detected clusters
    {
        find the cluster centers Cmby taking the mean
        find the total number of points in each cluster
    }
    If m>k
    {
        # Join two or more as follows
        select two cluster based on density and
        number of points satisfying the application criteria
        and joint them and find the new cluster center and
        repeat it until achieving k clusters.
        Finally we will have Ck centers
    }
    else {
        l=k-m
        # split one or more as follows
        if ( m >= l )
        {
            select a cluster based on density and number of
            points satisfying the application criteria and split it
            using kmeans clustering algorithm and repeat it
            until achieving k clusters.
            Finally we will have Ck centers
        }
    }
}
```

International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati
Steps
- Initially we apply the dbscan then we have m clusters.
- Then again we apply the kmeans taking k as input.
- If k<m then the m is reduced to m by clubbing the clusters
- If k>m then the m is partitioned to k clusters.

V. CONCLUSION

In this project we have successfully implemented the DBKMEANS which is the improved version of the kmeans. In this project we have taken many considerations to improve the kmeans such as the taking the means as the initial means & taking the silhouette width etc. We got the good results by using the DBKMEANS than the kmeans & the dbscan. We also showed the difference using the visualization. Finally by using this algorithm we overcome the many disadvantages in the kmeans & some disadvantages in the dbscan.

ACKNOWLEDGEMENT

This research is supported by the vignan university, AndhraPradesh, india. Mr. Krishna kishore, HOD and assistance professor Mrs. P. jhanasi lakshmi.

REFERENCES

[2] Raymond T. Ng and Jiawei Han, CLARANS: A Method for Clustering Objects for Spatial Data Mining, IEEE TRANSACTIONS ON KNOWLEDGE and DATA ENGINEERING, Vol. 14, No. 5, SEPTEMBER 2002.
An Efficient Algorithm for Person Identification
using Phase components of Iris Images

A.G. Korke & B.F. Momin
Department of Computer Science and Engineering, Walchand College of Engineering, Sangli
E-mail : korkeashok@gmail.com, bfnomin@yahoo.com

Abstract – This paper explores an efficient algorithm for person identification using phase-components of iris image for matching. Our proposed technique uses the phase only correlation (POC) and Band limited phase only correlation (BLPOC) for matching iris images to identify person. In this proposed algorithm, the use of phase components in 2D (two dimensional) discrete Fourier transform of iris images makes possible to achieve highly robust iris recognition in a unified fashion with a simple matching algorithm. Experimental evaluation using an iris image database clearly demonstrates efficient matching performance of the proposed algorithm.

I. INTRODUCTION
Iris recognition, a biometric, provides one of the most secure methods of authentication and identification, as they are difficult to replicate and steal. Biometric identification utilizes physiological and behavioral characteristics to authenticate a person’s identity. The iris has many features that can be used to distinguish one iris from another. One of the primary visible characteristic is the trabecular meshwork, a tissue which gives the appearance of dividing the iris in a radial fashion[1].

Iris recognition is one of the most promising approaches due to its high reliability for personal identification. The various methods for recognition of human iris pattern include Daugman’s approach, Wilde’s approach, hamming distance, etc. In this paper we use phase components in 2D Discrete Fourier Transforms of iris images, makes possible to achieve highly robust iris recognition in a unified fashion with a simple matching algorithm. The details and the results of proposed algorithm are presented in this paper.

II. PREPROCESSING STAGE
The preprocessing step is designed to remove irrelevant parts correctly from the given image and to extract only the iris region.

---

International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati
An Efficient Algorithm for Person Identification using Phase components of Iris Images

International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati

eye image. Canny edge detection algorithm gives the edge points by doing following operation on the given eye image.

1. Applying Gaussian filter
2. Finding gradient
3. Non-maximum suppression
4. Thresholding

On these edge images we are applying circular Hough transform to identify the circular region from the eye image. We will see above steps in brief

1. Applying Gaussian filter:

In this step the image is smoothened using Gaussian Filter which results in blurring of the image and reducing the effects of noise. Smoothing suppresses noise or other small fluctuations in image. The Gaussian distribution in 2 dimensional forms is

\[ G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \]

Where \( \sigma \) Standard deviation

Once we get the Gaussian filter we apply this filter on input eye image and perform simple convolution operation.

2. Finding Gradients:

The Canny algorithm basically finds edges where the gray scale intensity of the image changes the most. This step finds the edge strength by taking the gradient of the image. The direction of the edge is computed using the gradient in the x and y directions.

\[ \theta = \arctan(G_y/G_x) \]

\( G_y \text{ and } G_x \) are the gradients in the y and x directions respectively. Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image.

3. Non-maximum suppression:

After the edge directions are known, non-maximum suppression now has to be applied. Non-maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value that is not considered to be an edge. This will give a thin line in the output image.

4. Thresholding:

The non-maximum suppressed image may contain many false edges caused by noise during acquisition of image. If a single threshold, \( T_1 \) is applied to an image, and an edge has an average strength equal to \( T_1 \), then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than \( T_1 \) is presumed to be an edge pixel. Then, any pixels that are connected to this edge pixel and that have a value greater than \( T_2 \) are also selected as edge pixels. Output of this step is shown in fig 2.

5. Finding a circle:

Hough transform is standard method for detection of shapes such as circles which are already parameterized from known Formulae. After performing a circular Hough transform on the threshold images from above step, the maximum value in Hough space corresponds to the center and radius of the circle. For this, an edge map is generated by calculating the first derivative of intensity values in an eye image and then thresholding the result. From the edge map, votes are cast in Hough space for the parameters of circles passing through each edge point. These parameters are the center coordinates x and y and the radius r, which are able to define any circle. According to the equation,

\[ r^2 = x^2 + y^2 \]

B. Iris Normalization and Eyelid Masking

This step is to normalize iris to compensate for the elastic deformation in iris texture. We unwrap the iris region to a normalized rectangular block of a fixed size (256×128). In order to remove the iris region

Fig 2: Output of canny algorithm

Fig 3: Iris Localization
occluded by the upper eyelid and eyelashes, we use only the lower half of the iris region shown in Fig. 3 and apply a polar coordinate transformation to obtain thenormalized image shown in Fig. 4. In the transformed iris image, the irrelevant eyelid regions should be masked as shown in Fig. 4

\[ I(x(r, \theta), y(r, \theta)) = I(r, \theta) \]
\[ x(r, \theta) = (1 - r)x_p(\theta) + rx_i \]
\[ y(r, \theta) = (1 - r)y_p(\theta) + ry_i \]

Where,
\( x_p, y_p \) and \( x_i, y_i \) are co-ordinates of pupil and iris boundary.

![Fig 4: Normalized iris with eyelid masked](image)

C. Contrast Enhancement

In some situations, the normalized iris image has low contrast. To enhance contrast we are using local cumulative histogram equalization technique [2]. In our algorithm, we transform the pixel value by using the local cumulative histogram evaluated within a small image block (of size 15 * 15 pixels) centered at the pixel to be converted.

III. MATCHING STAGE

A] Baseline Algorithm:

This section describes details about the matching processes, where following steps need to be performed.

- Preprocessed Image
- Preprocessed Image
- Effective Region Extraction
- Displacement Alignment
- Matching Score Calculation
- Matching Score

![Fig 5: Proposed Matching Algorithm](image)

This technique uses the phase components in 2D Discrete Fourier Transforms (DFTs) of given images. Before discussing the details of matching algorithm we will introduce a basic principle of phase-based image matching using the Phase-Only Correlation (POC) and Band-Limited Phase-Only correlation (BLPOC) functions [3], [4], [5].

Consider two \( N_1 \times N_2 \) pixels images \( x(n_1, n_2) \) and \( y(n_1, n_2) \) where assume that the index ranges are \( n_1 = -M_1, ..., M_1 \) \((M_1 > 0)\) and \( n_2 = -M_2, ..., M_2 \) \((M_2 > 0)\) for mathematical simplicity, and hence \( N_1 = 2M_1 + 1 \) and \( N_2 = 2M_2 + 1 \). Let \( X(k_1, k_2) \) and \( Y(k_1, k_2) \) denote the 2D DFTs of the two images, the 2D DFT of images \( x(n_1, n_2) \) is given by,

\[
X(k_1, k_2) = \sum_{n_1, n_2} x(n_1, n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} = A_x(k_1, k_2) e^{j\theta_x(k_1, k_2)} \quad (1)
\]

Where, \( A_x(k_1, k_2) \) is the amplitude and \( \theta_x(k_1, k_2) \) is the phase. In the same way, DFT of second image can be calculated. The cross-phase spectrum \( R_{xy}(k_1, k_2) \) is given by,

\[
R_{xy}(k_1, k_2) = \frac{X(k_1, k_2) Y^*(k_1, k_2)}{|X(k_1, k_2) Y^*(k_1, k_2)|} = e^{j\theta_x(k_1, k_2)} \quad (2)
\]

Where \( k_1 = -M_1, ..., M_1 \) and \( k_2 = -M_2, ..., M_2 \). \( R_{xy}(k_1, k_2) \) is the complex conjugate of \( Y(k_1, k_2) \) and \( \theta_x(k_1, k_2) \) denotes the phase difference of \( X(k_1, k_2) \) and \( Y(k_1, k_2) \). The POC function \( r_{xy}(n_1, n_2) \) is the 2D Inverse DFT of \( R_{xy}(k_1, k_2) \) given by,

\[
r_{xy}(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1, k_2} R_{xy}(k_1, k_2) W_{N_1}^{-k_1 n_1} W_{N_2}^{-k_2 n_2} \quad (3)
\]

The observation shows that the 2D-DFT of normalized iris image contains meaningless phase components in high frequency domain. To evaluate the similarity using the inherent frequency band within iristextures, BLPOC function was employed. Assume that the ranges of the significant frequency band are \( k_1 = -k_1, ..., k_1 \) \((0 \leq k_1 \leq M_1)\) and \( k_2 = -k_2, ..., k_2 \) \((0 \leq k_2 \leq M_2)\). Thus, the effective size of frequency spectrum is given by \( I_1 = 2k_1 + 1 \) and \( I_2 = 2k_2 + 1 \). The BLPOC function is given by,

\[
r_{xy}^{k_1 k_2}(n_1, n_2) = \frac{1}{I_1 I_2} \sum_{k_1, k_2} R_{xy}(k_1, k_2) W_{I_1}^{-k_1 n_1} W_{I_2}^{-k_2 n_2} \quad (4)
\]
If two images belong to the same individual, the BLPOC function gives a distinct sharp peak otherwise the peak drops significantly. The height of the peak gives a good similarity measure for image matching.

1. **Effective Region Extraction**

In this step, given pair of normalized iris images to be compared. Here we are supposed to eliminate irrelevant regions such as a masked eyelid and specular reflections. The purpose of this process is to extract effective regions of the same size from the two images, as illustrated in Fig.6.

![Fig. 6: Effective region extraction](image)

2. **Translational Displacement**

This step is to align the translational displacement between the extracted images. Rotation of the camera, head tilt, and rotation of the eye within the eye socket may cause the displacement in normalized images. The displacement parameters can be obtained as the peak location of the POC function $r_x \cdot r_y(n_1, n_2)$. Then obtained parameters are used to align the images.

3. **Matching Score Calculation**

In this step, we calculate the BLPOC function $r_x \cdot r_y(n_1, n_2)$ between the aligned images $x(n_1, n_2)$ and $y(n_1, n_2)$ and evaluate the matching score. In the case of genuine matching, if the displacement between the two images is aligned, the correlation peak of the BLPOC function should appear at the origin $(n_1, n_2) = (0, 0)$. So, we calculate the matching score as the maximum peak value of the BLPOC function.

**B. Modified Algorithm for Degraded Images**

This algorithm performs some modifications on the baseline matching algorithm which are suitable for degraded iris images. The baseline algorithm described in the previous part, performs image matching by using the whole iris image. Here in the modified algorithm, we divide the normalized iris image into overlapped blocks of size 128*64 where we are getting total 9 image blocks of equal size. We compute the BLPOC function for every block pair. Then, we take an average of the computed BLPOC functions across the whole image plane to improve the peak-to-noise ratio of the correlation surface. This technique leads to better discrimination capability, even for highly degraded iris images.

**IV. RESULT ANALYSIS**

This section describes a set of experiments using the CASIA iris image database (ver. 1.0) [6] for evaluating matching performance. We first evaluate the genuine (intra-class) matching scores for all the possible combinations of genuine attempts $(7C2 \times 10^4 = 2184$ attempts). Next, we evaluate the impostor (inter-class) matching scores, where we take a single image for each eye and make all the possible combinations of impostor attempts. Figure 7 shows distributions of genuine and impostor matching scores. The figure shows a good separation of genuine and impostor matching scores, where the minimum genuine matching score is 0.13, and the maximum impostor matching score is 0.12. With these matching scores, we obtained 1.79% False acceptance rate for baseline algorithm and 0.4% False acceptance rate for modified algorithm. The result demonstrates a potential possibility of phase-based image matching for creating an efficient iris recognition system.

**V. CONCLUSION**

An approach based on phase component for image matching using POC and BLPOC functions was successfully implemented for iris recognition. The use of the Fourier phase spectra of iris images makes it possible to achieve highly accurate iris recognition with a simple matching algorithm. The experimentation on these implemented algorithms indicated a superior capability of BLPOC function over POC function for recognition. A matching score
An Efficient Algorithm for Person Identification using Phase components of Iris Images

of 0.13 shows a false acceptance rate of only 1.79% for baseline algorithm and 0.4% for modified algorithm indicating usefulness of image matching using phase components for iris recognition.

REFERENCES


Verification of C/C++ IP Models using RTL vcd files and STL files to Accelerate Product Time to Market

Basavaraju.G¹ & Shankaraiah²

¹ Intel mobile communications (IMC), Bangalore, India
² Sree Jayachamarajendra College of Engineering, Mysore, India
E-mail : Basavaraju.g@intel.com¹, shankarsjce@gmail.com²

Abstract – Virtual Prototypes (VPs) have become essential part of the embedded software development. High level models of IPs (intellectual property) are needed to create the VPs. Development and verification effort for these high level models plays vital role in the timely availability of the VPs. Verification of the high level models is a major challenge in terms of both effort and to prove equivalence with its RTL counterpart. Re-use of RTL test benches presents one option to reduce the verification effort; this however, has several tooling and interoperability issues. An approach to use trace information like Value Change Dump (VCD) as input to generate test vectors for high level models alleviates these limitations, and several authors have successfully explored this approach. While this approach enables the use of trace information as-is it does not allow for modification/updates to test vectors; which may be necessary to suit to the high level models. This paper presents as approach which apart from enabling as-is use of trace information, provides the flexibility to modify or enhance the test vector where needed. And this paper also introduces new way of verifying the high level models using socket transaction language (STL) files. This approach blends the re-use of test vectors from RTL traces with hand crafted test cases specifically developed for high level models resulting in reduced verification effort and enabling early availability.

Keywords - functional verification, high level modeling, virtual prototyping, simulations.

I. INTRODUCTION

Virtual prototyping (VP) [10] is an emerging technology, which is simulation of the physical prototype in virtual environment. VP refers to the analysis of component without actually making a physical prototype of the part; it is a computer model (Virtual prototype) which allows the user for observation, analysis and manipulation. High level models (C, C++, and Java) are typically in software, or software like language. Major goal of High level model is to enable verification at higher level of abstraction, enabling early exploration of system level designs. Increasing the complexity is forcing design to move above the Register transfer language (RTL). These High Level Models are normally verified by creating direct tests and then outputs are compared manually or visually to establish cycle accuracy. This is not suitable for timing testing of non-trivial designs. A High Level Model allows the modeling of Hardware IP at higher abstraction level than RTL this makes reusing the RTL Trace information. There are two ways to verify High level models with RTL information by reusing the RTL Test bench and another way is co-simulating RTL test bench with High level models. Creation of stimulus and test bench for performance and performance validation is time and resource intensive proposition. Number of tools available for verifying HDL design. For SystemC [5] designs verification tools are only starting to appear. Most of the IP existing today are RTL (Register transfer language) IP’s. The idea of Re-using a RTL trace file is already explored by many Authors. Amit Nene [4] introduced Trace Based Verification Methodology for C/C++ IP Models and also publicized that component validation execution flow is much faster than the older methods. This approach is scalable to get to a regression setup for any component. In addition, the confidence in transaction level modeling (TLM) [8] users is highly enhanced. However these approaches are valid only when RTL trace file is available, and user does not have flexibility to reuse, modify and create test cases. In this paper, a flexible approach is proposed for the user to reuse, modify the test cases. Proposed method could also be used to functionally verify the model even though RTL trace file is not available. the advantages of this approach, including no extra tools required, cost is minimized due to re-use of test benches developed earlier and we can save some time on test bench effort for High level models. This approach also involves in generating socket transaction language (STL) [6] file...
Verification of C/C++ IP Models using RTL vcd files and STL files to Accelerate Product Time to Market

from the RTL trace file. This paper also explains about verifying an IP at transaction level. The term High level model refers to the software model typically written in C or C++ and the term transaction refers to the exchange of data or event between two components.

II. WHAT IS SYSTEMC?

SystemC [5] is set of C++ class definitions, Macros and a Methodology for using these classes. It is a system-level modeling language based on C++. SystemC uses an object-oriented approach to achieve abstraction, modularity, compositionality, and reuse. The base layer of SystemC provides an event-driven simulation kernel. This kernel operates at the event level switches execution between processes. The basic building block in systemc is the module. A module is container that contains one or more processes to describe the parallel behavior of the design. A module can also contain other modules, representing the hierarchical nature of the design. Processes execute concurrently; the code within each process executes sequentially. Processes inside a module communicate via signals. Modules communicate via channels. The channels are abstract and are accessed via their interface methods. Modules have ports that are bound to interface methods. More Details of SystemC and language constructs are given in [5].

III. VERIFICATION OF HIGH LEVEL MODELS

A. Traditional Verification Method

Typical approach for verifying high level models like SystemC TLM models relies on having system with a stimuli generator and checker connected to the design under test. Stimuli generator uses threads to write to ports and verify the IP.

B. Verification of High Level Models using RTL Test bench

In a typical scenario, test-benches are developed individually for higher level simulations like SystemC simulations as well as for lower level like RTL. However RTL tests and TLM DUT can be co-verified with appropriate tooling involved. This approach relies on a tool which allows mixed language simulation between SystemC and HDL. This would nullify the effort involved in developing SystemC test-benches.

C. Proposed approach: Verification of High Level Models using RTL Trace

In this proposed approach, the trace from RTL simulations is used to verify the TLM models. It is expected that if the same stimulus is applied to both RTL and TLM the response should remain same. Firstly the RTL model is simulated using test stimuli. The traces on interface signals of RTL model are captured to record the sequence of the RTL interface events in VCD [3] format. These sequences are then converted to transactions over TLM model.

The VCD file holds information of value changes of signals along with timestamp. VCD files are post processed to create high level test vectors. RTL interface/communication is at pin/signal level while TLM models communicate in terms of structures. Group of signals together make a logical entity which can be used as a transfer unit in the TLM models; this would be referred to as payload. Firstly the RTL tests are run on the design to generate VCD file. The generated VCD file is provided as an input to the VCDParser. VCDParser (i.e script) reads the VCD file line by line and determines which signals have changed and at what timestamp. VCDParser stores the identified signal name and their values in data structure called associate container (associate container is a container which stores the data in a non-liner fashion as key-value pair) along
with time stamps. This associated container is referred as a Signal Data Base (SDB).

![Table](image)

<table>
<thead>
<tr>
<th>Time stamp</th>
<th>Signal name</th>
<th>Signal value</th>
<th>Signal name</th>
<th>Signal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>address</td>
<td>1</td>
<td>command</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>result</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3 : Architecture of SDB

Above procedure is repeated till VCD file ends. Since timestamp is unique, in SDB timestamp is considered as key and signal name and signal values are stored as value pair. It is suggested to use lookup table instead of using iterative structures for faster access of SDB information. Time complexity of lookup table is $O(1)$.

![Diagram](image)

Fig. 4 : VCD file Parser generating cases.

The transaction generator block in VCDParser processes the SDB information and generates TLM transactions and signal reads & writes (i.e. test case file). The interface file is taken as input to identify the input and output ports for test case file generation.

Sample test case file generated from VCDparser:

```cpp
wait(100,sc_ps);
address=1000;
data=reading_from(address);
cout<<"READING .."<<sc_time_stamp()<<endl;
cout<<address<<"==>":"<<data<<endl;
wait(100,sc_ps);
data=abc;
address=2000;
writing_to(address,data);
cout<<"writing to.."<<address
<<sc_time_stamp()<<endl;
```

**VCDParser within SystemC Test Bench:** In this method VCDParser is part of the SystemC test bench. Below diagram depicts complete setup.

![Diagram](image)

Each of the components is described below.

- **SystemC Test bench:** Top-level component that instantiates and collaborates with other components.
- **VCDParser:** Component responsible for parsing VCD file, update the SDB (signal data base), generate transaction payload and TLM transactions.
- **Input Driver:** Input drives the transactions on the DUT (Design under Test) interface.
- **SystemC DUT:** High level model to be verified.

The simulation involves in

- Running the RTL simulation and collecting the VCD traces
- VCD trace is provided as input to VCDParser.
- VCD parser generates the transaction payloads at associated timestamps.
- Input driver drives these payloads at the associated timestamps on the DUT interface.
- DUT simulates and generates the VCD traces.

Post simulation, it would involve manual effort to compare output trace of High Level Model with the RTL recorded sequences to decide the status of the test (PASS/FAIL).

If the VCDParser (i.e. script) is placed within SystemC test bench, it makes very difficult to debug the code in case of errors and also VCDParser parses only one VCD file in single simulation. This restricted the user to use only one VCD file in simulation. Normally DUT functionality is verified in two ways

1. Directly writing test cases in the SystemC test bench.
2. Generating the test cases from RTL VCD file then placing these test cases in SystemC test bench.

With this approach DUT functionality is verified only when the RTL VCD file is available. This framework doesn’t provide flexibility for the designer to write his own test cases in SystemC test bench.

Briefly describing the drawbacks of this approach:

- Increases the complexity of understanding the test cases in debugging.
- Dependency on RTL vcd files availability for Model verification.
- Writing new test-cases is not possible.
- Execution of multiple VCD files in a single simulation is not possible.

**VCDParser outside SystemC Test Bench:** With this approach, the VCDParser is separated out from the test-bench which means that VCDParser (i.e script) is executed independent of SystemC test bench. The VCD parser includes test-case file generation in a compilable file format which initiates SystemC TLM transactions. These file are included for test-bench compilation and are executed for DUT verification. Following diagram depicts the flow.

Separating out the VCDParser from the test-bench helps in sorting out most of the drawbacks listed when VCDParser being part of test-bench. The main advantage this method is, VCDParser is used when RTL vcd file available. If VCD file is not available, then designer can be able write his own test cases.

With proposed approach it would be possible for

- Writing new test-cases is possible.
- Execution of multiple VCD files in a single simulation is possible
- Reuse of test cases is possible.

**High Level Model Verification with STL files:** The above framework (i.e VCDParser) is enhanced to provide test cases as STL (Socket Transaction Language) [7] in a file format because Test case files generated from the VCDParser is not in the standard format. It makes very difficult for others to understand the test case.

STL is standard format and it is scripting language for generating Q-Masters test vectors to generate and control the open core protocol (OCP) [2] input traffic. STL is a standard defined by OCP-IP [8]. These vectors are parsed using STL Parser modules like Generic File Reader Bus Master (GFRBM) [10] and verify DUT. Following diagram depicts the flow.

The simulation involves

- Running the RTL simulation and collecting the VCD traces
- VCD trace is provided as input to VCDParser.
- VCD parser generates the test cases as STL vectors in file format
- STL Parser/GFRBM parses and drives transactions at the associated timestamps on the DUT interface
- DUT simulates and generates the VCD traces.

Post simulation, it would involve manual effort to compare output trace of High Level Model with the RTL recorded sequences to decide the status of the test.

Below is a Sample STL file:

Signal flags=1 (3)

Idle 103637000
0xread16 0x080 0x2

Idle 40313000
0xwrite16 0x08c 0x2

Advantages of STL file generation:

- STL file is a standard format. Any vendor models like Generic File Reader Bus Master (GFRBM) which supports STL parsing to TLM transactions or
internally developed models can be used as testbench for DUT verification.

- STL has a standard commands [6] to write test cases in case of RTL VCD file not available.
- STL syntax makes Easy for others to understand and debug the STL files

VCDParser supporting both Transactions generation and STL vector generation:

It is suggested, that the VCDParser supports both Transaction generation and STL vector generation in files. It would be then the user choice for

1. Use the Transaction generated file for test-bench compilation and verify DUT or
2. Use STL Vectors and STL Parser for DUT Verification.

Below diagram describes top level view of proposed approach.

![Diagram](image)

**Fig. 8:** VCDParser supporting both test case and STL file generation

**IV. WORK DONE**

Proposed methodology is evaluated with timer module. RTL timer model is simulated for some of the tests and respective VCD dump is captured. This VCD file is provided as input to VCDParser for STL and Transaction file generation. The Software model of Timer Model is successfully verified with both of the proposed approaches. The generated trace from high level model and VCD dump of RTL model are compared for functional correctness. Time stamp captured in the high level model vcd file doesn’t match exactly with that of RTL vcd file. This is due to non ideal conditions of software model. It involves some manual effort to validate the functional correctness of the high level model with these time stamp differences. The time taken to write test cases to verify high level model with RTL VCD FILE approach and without RTL VCD FILE approach is given in table 1.

**TABLE I : TIME TAKEN TO WRITE TEST CASES**

<table>
<thead>
<tr>
<th>Module name</th>
<th>Without VCD file</th>
<th>with VCD file</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMER MODULE</td>
<td>65 days</td>
<td>10 days</td>
</tr>
<tr>
<td>CRYPTOGRAPHIC MODULE</td>
<td>80 days</td>
<td>15 days</td>
</tr>
</tbody>
</table>

**V. CONCLUSION**

With proposed approaches, one can save considerable amount of effort spent in writing test cases for High level models. This in turn improves the Component validation flow compared to the traditional, co-simulation verification flow. The high level models are verified with RTL trace in standalone manner with no dependency on licenses or tools. Using STLparser high level models can be verified even if RTL traces are not available.

**REFERENCES**


Reducing the False Positives in intrusion Detection using Classification Algorithm

R. Ravinder Reddy, Y.Rama Devi, K.V.N Sunitha & A. Sirisha
Department of CSE, CBIT, GNIT
E-mail : ravi.ramasani@gmail.com, yrdcse.cbit@gmail.com, k.v.n.sunitha@gmail.com

Abstract – Intrusion Detection Systems (IDSs) are used to monitor computer systems for signs of security violations. Since the first intrusion detection system and up to this moment all IDSs had generated thousands and thousands of alerts and most of these alerts are false alerts, which lead the researchers to develop an idea to reduce the rate of the alerts or at least the false alerts of them. One of the ideas was to use classification algorithms it will classify the alerts accurately and by using machine learning algorithms to reduce the false alerts rate. Continuous monitoring of all types alerts and there by evolving a judgment for improving security is the major concern in the proposed model.

Keywords - False positives, intrusion detection, classification, computer security

I. INTRODUCTION

Several methodologies have been applied to resolve the problem of false positives. Internet, mobile technologies, Computers became part of day-to-day life. As reliance on connectivity for computing and sharing of data is mandatory, computers, storage devices and mobile devices are connected to Internet. The IT installations, confidential data are susceptible to cyber attacks. False Positives will occur, if stringent rules are enabled to increase security to reduce false negatives. False positives while customizing to environment with additional functionality.

Network-based intrusion detection systems (NIDS) perform in-depth packet analysis in order to enumerate attackers who are attempting to expose network and service vulnerabilities. NIDS devices can also aid in identifying misuse patterns and gathering forensic data. By examining network traffic in real time, NIDS devices can alert users to possible attacks and/or take predefined responsive actions to help mitigate the threat. By providing an additional layer of protection above and beyond access control devices such as a firewall, NIDSs can be a valuable addition to the security arsenal. However, network intrusion detection has been criticized for its propensity to generate a perceived large amount of false positives and false negatives. Effective NIDS device management can appreciably reduce these reporting inaccuracies.

II. FALSE POSITIVES OR FALSE ALARMS

The term false positive is a broad and somewhat vague term that describes a situation in which an NIDS device trigger an alarm in a when there is malicious activity or attack occurring. Other common terms used to describe this condition are "false alarms" and "benign trigger". False alarm is the better term to describe this behavior since "false positive" gives the impression that IDS technology itself is fundamentally flawed and benign trigger gives the impression that there is no possibility for a true false positive to exist. Here I will use the term false alarm to describe the general condition of an alarm being generated without a true security related event. False alarms are the Internet security equivalents of the boy who cried wolf. They are problematic because by triggering unjustified alerts, they diminish the value and urgency of real alerts.

The open source Snort is used to experimenting. In snort IDS, threat identification is based on signatures and rules or anomalies of each and every attack and their variants. Hence, each attack, and its variants is identified, analyzed, signature will be prepared and deployed. If the process of remediation is uncoordinated among multiple Variant signatures, and creates an ever-increasing number of signatures, producing more alerts, may be false positives. Like industry standard IDS, Snort also able to identify malicious code by Common Vulnerabilities and Exposures (CVE) ID [4] or other equivalent identifier. IDS generally do not have the intelligence to determine, if any of the machines on the
network are susceptible to an attack with CVE ID, or if any machine has a specific vulnerability to that attack, or if targeted vulnerability has already been patched. IDS have been studied for more than 28 years since Anderson’s [5] (1980) report. It is concept that an intruder’s behavior will be different from that of a legitimate user and that actions can be detectable. As per Peng Ning[6] (2005) Intrusion detection systems (IDSs) are usually deployed along with other preventive security mechanisms, such as access control and authentication, as a second line of defense that protects information systems.

IDS architecture consists of various layers. The traffic (Packets) is acquired from the network are passed through layers of decoder routines. Each preprocessor checks and then sent through the detection engine. The detection engine checks each packet against various options listed in the Snort rules files. Content, flow, flow-bits, pcre, byte-test, byte-jump are the functions for matching.

The network IDS will do packet analysis to detect the intrusions. The first process is decoding each packet. The decoder will identify the protocols used in the packet like TCP, ICMP, Ethernet, UDP, IP. The Protocol information and the location of the payload in the packet, and size of payload will be saved for use by preprocessor and detection engines. If configured, the decoder opens packet headers, examine for errors or anomalies in header fields and output an alert or drop the packet. For example, the protocol field indicates IPv4 and the size is less than 20 bytes, the minimum length for an IPv4 header, IDS will generate an alert and drop the packet, with alert. In Snort IDS, “Performance Monitor preprocessor measure performance and turns on event reporting and prints out statistics as to the number of signatures that were matched by the set wise pattern matcher (non-qualified events) and the number of those matches that were verified with the signature flags (qualified events). This shows the user if there is a problem with the rule set that they are running”.

FALSE NEGATIVES (FN):

False negative is the term used to describe a network intrusion device’s inability to detect true security events under certain circumstances. In other words, malicious activity is not detected and alerted. Fortunately, there are actions that can be taken to reduce the chance of false negative conditions without increasing the number of false positives. The difficulty in creating this "balance" is to create a more manageable NIDS deployment without introducing extra risk. First, however, we need to analyze how network intrusion detection systems detect these attacks so we can understand the consequences associated with our actions.

FALSE POSITIVES (FP):

False Positives usually derive from causes that may be related to the topology of the network, misconfigured hosts or periodical nominal services and tasks that are carried out in the network. All these causes are time invariant and produce recurrent patterns of FPs.

III. PROPOSED MODEL

The main reasons for the wrong interpretation is the rules are configured so,The behavior is not analyzed the stimulus and responses and the accuracy of the analyzed data models and rules. In this we used classification algorithm to divide properly in between false positives and false negative. We achieved the 99.987% accuracy.

Need for Accuracy:

The rate of false positives and false negatives compared to actual attacks—the noise-to-signal ratio—should be very low. False positives are false alarms; alarms that are set off even when there is no attack. If the number of false positives is very high, the produced information becomes unreliable, and real incidents might go undetected. False positives are typically generated by systems that rely on a single detection method, and by ones that cannot be configured at different levels to fit into the operational environment. An additional reason for false positives is that the first-generation intrusion detection products do not offer any means of correlating events. False negatives, on the other hand, are missed attacks. The IPS needs to be able to detect each and every real attack.

In the classification algorithm we solved the problem through a combination of powerful detection methods that are applied according to administrator-defined rules, and by introducing intelligent event correlation. In addition, there are several improvements in the detection methods themselves, such as context-sensitive, regular expression-based fingerprints, and configurable protocol inspection modules. This way, real threats are acted upon more readily, and less manual work is wasted on analyzing false alarms.

IV. EXPERIMENTAL WORK

In this we used the c4.5 classification algorithm to prune tree and obtained the proper results in classifying between false positives and false negatives accurately. We used the decorate method for implementing it. The decorate is a meta-learner for building diverse ensembles of classifiers by using specially constructed artificial training examples. Comprehensive experiments have demonstrated that this technique is consistently
more accurate than the base classifier, Bagging and Random Forests. Decorate also obtains higher accuracy than Boosting on small training sets, and achieves comparable performance on larger training sets.

The performance of classification systems is usually evaluated using ROC analysis (Fawcett, 2003). The performance of such a system can be expressed by the true positive rate and the false positive rate:

--- Run information ---

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: packet
Instances: 8119
Attributes: 13

| duration <= 0: normal (46.0/4.0) |
| duration > 0 |
| duration <= 6: probe (12.0/4.0) |
| duration > 6: normal (2.0) |
| src_bytes > 0: normal (552.0/6.0) |
| count > 5: DOS (95.0/1.0) |
flag = RSTR
| count <= 8: normal (14.0) |
| count > 8: probe (9.0) |
flag = S1: normal (1.0)
flag = REJ

Test mode: evaluate on training data

--- Classifier model (full training set) ---

J48 pruned tree

-------------

flag = SF
| count <= 0: DOS (1736.0)
| count > 0
| count <= 5
| src_bytes <= 0
| dst_bytes <= 1: probe (21.0)
| dst_bytes > 1

service = private: probe (1.0)
service = domain_u: probe (0.0)
service = http: probe (0.0)
service = HTTP: probe (0.0)
service = smtp: probe (0.0)
service = ftp_data: probe (0.0)
src_bytes <= 0
| service = ftp: probe (0.0)
| service = eco_i: probe (0.0)
| service = auth: probe (0.0)
| service = ecr_i: probe (0.0)
| service = IRC: probe (0.0)
| service = X11: normal (7.0/1.0)
flag = RSTR
| service = X11: probe (0.0)
| service = telnet: probe (0.0)
| service = domain: probe (2.0)
flag = S1: normal (1.0)
flag = REJ
| service = pop_3: probe (0.0)
| service = Idap: probe (0.0)
| service = login: probe (0.0)
| service = name: probe (14.0)
| service = ntp_u: probe (0.0)
| service = http_443: probe (0.0)
| service = sunrpc: probe (0.0)
Reducing the False Positives in intrusion Detection using Classification Algorithm

| service = printer: probe (0.0) | service = shell: probe (0.0) |
| service = systat: probe (1.0) | service = efs: probe (0.0) |
| service = tim_i: probe (0.0) | service = klogin: probe (0.0) |
| service = netstat: probe (1.0) | service = kshell: probe (0.0) |
| service = remote_job: probe (14.0) | service = icmp: probe (0.0) |
| service = link: probe (13.0) | flag = S3: DOS (0.0) |
| service = urp_i: probe (0.0) | flag = RSTO: probe (5.0) |
| service = sql_net: probe (0.0) | flag = S0 |
| service = bgp: probe (0.0) | count <= 83 |
| service = pop_2: probe (0.0) | service = private: DOS (0.0) |
| service = tfp_u: probe (0.0) | service = domain_u: DOS (0.0) |
| service = uucp: probe (0.0) | service = http: DOS (0.0) |
| service = imap4: probe (0.0) | service = HTTP: DOS (0.0) |
| service = pm_dump: probe (0.0) | service = smtp: DOS (0.0) |
| service = nsp: probe (0.0) | service = ftp_data: DOS (0.0) |
| service = courier: probe (0.0) | service = ftp: DOS (0.0) |
| service = daytime: probe (1.0) | service = eco_i: DOS (0.0) |
| service = iso_tsap: probe (0.0) | service = other: DOS (0.0) |
| service = echo: probe (1.0) | service = auth: DOS (0.0) |
| service = discard: probe (1.0) | service = ecr_i: DOS (0.0) |
| service = ssh: probe (1.0) | service = IRC: DOS (0.0) |
| service = whois: probe (14.0) | service = XLL: DOS (0.0) |
| service = mtsp: probe (14.0) | service = x11: DOS (0.0) |
| service = gopher: probe (3.0) | service = xll: DOS (0.0) |
| service = rje: probe (14.0) | service = finger: DOS (0.0) |
| service = ctf: probe (13.0) | service = time: DOS (0.0) |
| service = supdup: probe (0.0) | service = domain: DOS (0.0) |
| service = hostnames: probe (0.0) | service = telnet: DOS (15.0) |
| service = cns_net: probe (0.0) | service = pop_3: DOS (0.0) |
| service = uucp_path: probe (0.0) | service = ldap: DOS (0.0) |
| service = nntp: probe (1.0) | service = login: DOS (0.0) |
| service = ntim: probe (1.0) | service = name: DOS (0.0) |
| service = nethios_nt: probe (0.0) | service = ntp_u: DOS (0.0) |
| service = nethios_dgm: probe (0.0) | service = http_443: DOS (0.0) |
| service = netbios_sss: probe (0.0) | service = sunrpc: DOS (0.0) |
| service = vmnet: probe (0.0) | service = printer: DOS (19.0) |
| service = Z39_50: probe (0.0) | service = systat: DOS (0.0) |
| service = exec: probe (0.0) | |
### Service List

<table>
<thead>
<tr>
<th>Service</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>tim_i</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>netstat</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>remote_job</td>
<td>probe (2.0)</td>
</tr>
<tr>
<td>link</td>
<td>probe (3.0)</td>
</tr>
<tr>
<td>sql_net</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>bgp</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>pop_2</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>tftp_u</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>uucp</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>pm_dump</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>nntp</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>courier</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>daytime</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>iso_tsap</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>echo</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>discard</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>ssh</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>whois</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>mtp</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>gopher</td>
<td>probe (2.0)</td>
</tr>
<tr>
<td>rje</td>
<td>probe (3.0)</td>
</tr>
<tr>
<td>ctf</td>
<td>probe (2.0)</td>
</tr>
<tr>
<td>supdup</td>
<td>probe (3.0)</td>
</tr>
<tr>
<td>hostnames</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>csnet_ns</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>uucp_path</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>nntp</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>netbios_ns</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>netbios_dgm</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>netbios_ssn</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>vmnet</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>Z39_50</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>exec</td>
<td>DOS (8.0)</td>
</tr>
<tr>
<td>shell</td>
<td>DOS (20.0)</td>
</tr>
<tr>
<td>efs</td>
<td>DOS (8.0)</td>
</tr>
<tr>
<td>klogin</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>kshell</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>icmp</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>S2</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>SF</td>
<td>normal (1296.0/8.0)</td>
</tr>
<tr>
<td>S1</td>
<td>normal (8.0)</td>
</tr>
<tr>
<td>REJ</td>
<td>normal (0.0)</td>
</tr>
<tr>
<td>S3</td>
<td>normal (0.0)</td>
</tr>
<tr>
<td>RSTOS0</td>
<td>probe (16.0)</td>
</tr>
<tr>
<td>SH</td>
<td>probe (45.0/1.0)</td>
</tr>
<tr>
<td>OTH</td>
<td>DOS (0.0)</td>
</tr>
<tr>
<td>logged_in</td>
<td>1</td>
</tr>
</tbody>
</table>

---

Number of Leaves : 164
Size of the tree : 177

Time taken to build model: 0.38 seconds

### Summary

Correctly Classified Instances  8079   99.5073 %
Incorrectly Classified Instances  40   0.4927 %
Kappa statistic  0.9883
Mean absolute error  0.0037
Root mean squared error  0.0432
Relative absolute error  2.2137 %
Coverage of cases (0.95 level)  99.6305 %
Mean rel. region size (0.95 level)  20.1897 %

Total Number of Instances  8119
Reducing the False Positives in Intrusion Detection using Classification Algorithm

1. Check for base cases
2. For each attribute a
   1. Find the normalized information gain from splitting on a
3. Let \( a_{best} \) be the attribute with the highest normalized information gain
4. Create a decision node that splits on \( a_{best} \)
Recurse on the sublists obtained by splitting on \( a_{best} \), and add those nodes as children of node

V. CONCLUSION

It seems clear that there is no single device or method will solve all the network security concerns. A proper solution is to implement defense-in-depth, which means adding layers of defense to your network covering all the different sectors in the presented security model. Besides firewalls, network and host-based intrusion detection/protection, as well as antivirus software all contribute to enhanced network security. Despite all the promises however, the general view is that the traditional Intrusion Detection Systems have failed to meet expectations in many areas. More specifically, the traditional IDS systems are:

- Inaccurate with a high noise-to-signal ratio
- Too passive-detection alone is no longer sufficient
- Unable to meet the throughput and reliability requirements of today's networks.

There is no doubt that Intrusion Detection Systems are an important element of network security and that they bring value. At the same time, in order to survive, IDS systems need to meet the challenges outlined above—otherwise this vital technology faces death due to operations becoming too expensive in return for the benefit they provide.

REFERENCES


Reducing the False Positives in intrusion Detection using Classification Algorithm


Secure Multicast Routing Protocol Enabling Content Based Routing in MANET

Prajakta Agavekar & Medha A. Shah
Dept. of Computer Science & Engg., Walchand College of Engineering, Sangli, India
E-mail : agavekar.prajakta@gmail.com, shah.medha@gmail.com

Abstract – We present a Secure Modified Multicast Ad-hoc On-demand Distance Vector routing protocol enabling Content Based Routing in MANET. The protocol can organize the nodes of the MANET in self-repairing tree topology and also secure the nodes from Black Hole Attack. As Content-based routing (CBR) provides a powerful and flexible foundation for distributed applications, its communication model, based on implicit addressing, fosters decoupling among the communicating components. So meeting the needs of many dynamic scenarios including mobile ad hoc networks (MANETs) is very important. The application-level routers that provide CBR are organized in a tree-shaped network with a fixed topology. This structure cannot meet the characteristics of a CBR model. Also very few protocols in MANET ensure security of the network. The protocol discussed in this paper is 1) able to self-repair to tolerate the frequent topological changes in MANET, 2) minimize the topological reconfigurations by arranging the nodes in a tree topology, and 3) secure the nodes from black-hole attack.

Keywords - Content Based Routing, MAODV, Black Hole Attack, Publish-Subscribe Systems

I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a self-organizing adaptive network composed of a dynamic collection of wireless mobile devices that can communicate and move at the same time. MANETs can be formed and de-formed on-the-fly without the support of a centralized administration function and fixed wired infrastructures. The main issue in MANET is to provide the application layer with suitable communication abstractions that can fit the very dynamic nature of the underlying network. The issue can be resolved by using Content Based Routing (CBR). CBR differs from classical routing in that messages are addressed based on their content instead of their destination.

The unique characteristics of MANETs present a new set of nontrivial challenges to security design. Along with the advantages of Content Based Routing, security of the protocol is of utmost concern. Currently, the protocol doesn’t specify any special security measures, hence security must be provided at cost overhead, as route protocols are prime targets of attacks. Some of the possible attacks are black hole attack, wormhole attack, rushing attack, selfish nodes, etc.

Here we chose MAODV protocol as the basic protocol to modify so as to meet the aim of the protocol.

The protocol discussed in this paper focuses on all these aspects.

The rest of the paper is organized as follows. Section 2 gives overview of some important concepts. Section 3 describes the methodology to implement the protocol. Section 4 shows the results of simulation experiments under different scenarios. Finally, section 5 summarizes the conclusions.

II. OVERVIEW OF IMPORTANT CONCEPTS

A. Content Based Routing(CBR)

CBR provides a powerful and flexible foundation for distributed applications. Its communication model fosters decoupling among the communicating components, thus meets the needs of many dynamic scenarios including MANET. Publish-Subscribe systems are very good example of CBR. The publish/subscribe interaction scheme is receiving increasing attention and is claimed to provide the loosely coupled form of interaction required, between senders and receivers, in large scale networks. Subscribers have the ability to express their interest in an event, or a pattern of events, and are subsequently notified of any event, generated by a publisher, which matches their registered interest. An event is asynchronously propagated to all subscribers that registered interest in that given event. This decoupling between publisher and subscriber is obtained.
through the fact that publishers and subscribers do not know each other, and in particular the delivery of a message to all the interested subscribers, are realized by a dispatching service. Therefore, in CBR, it is the receiver that determines message delivery, not the sender.

CBR is not simply multicast though it enables multipoint communication. Multicast approaches typically propagate messages along a tree defined on a per-group basis, spanning all the receivers for that group. Therefore, a message addressed to multiple groups is typically duplicated at the sender, and each copy is routed independently. In CBR, the absence of an explicit and a priori definition of groups, along with the fact that every message can be addressed to a different set of components, discourages the use of separate per-group trees. The sharp decoupling induced by CBR form of communication enables one to easily add, remove, or change components at runtime with little impact on the overall architecture.

B. Black Hole Attack

A black hole attack is one in which a malicious node uses the routing protocol and advertises itself as having the shortest path to a destination in a network. This attack aims at modifying the routing protocol so that traffic flows through a specific node controlled by the attacker. The attacker drops the received messages instead of relaying them as the protocol requires. This can cause Denial of Service (DoS) by dropping the received packets. A black hole attack can be achieved by a single node or by several nodes. A single node black hole attack forges the sequence number and hop count of a routing message in order to forcibly acquire the route, and then eavesdrop or drop all data packets that pass. The reply from the malicious node reaches the source node earlier than the reply from the legitimate node, as the malicious node does not have to check its routing table as the other legitimate nodes. The source chooses the path provided by the malicious node and the data packets are dropped. Thus, malicious node forms a black hole in the network and this problem is called black hole problem.

C. Multicast Ad-hoc On-demand Distance Vector (MAODV)

MAODV is a multicast routing protocol for ad hoc networks that dynamically constructs a shared multicast tree which connects the group members possibly via some non-member nodes. As nodes join the group, a tree is created. This tree connects the group members and many routers which are not group members but exist in the tree to connect the group members. Multicast group membership is dynamic. The group members and routers are all members of the tree. Every multicast group is identified by a unique address and group sequence number for tracing the freshness of the group condition. When a node wants to find a route to a group or join a group, it broadcasts a RREQ message. Any node with fresh enough route to the multicast group may respond to this request message with a RREP message. If a node wants to become a member of the group that does not exist, then this node becomes the leader of that group and is responsible for maintaining the group. Group Hello messages are broadcasted periodically to check for connectivity of the tree structure. This results in increased overhead in maintaining routes.

III. METHODOLOGY

As MAODV doesn’t support CBR with dynamic tree topology, changes must be made into MAODV so as to achieve the purpose of CBR. In the following aspects, changes to MAODV are to be made.

1) Minimization of Route Changes: When a change in the broker tree occurs, the routing information used to perform CBR must be updated accordingly. To improve the overall performance, reply should be chosen that results in the shortest reconfiguration path. This information is not available in MAODV. To overcome this limitation, ancestor list containing the identifiers of all its tree ancestors can be maintained at each node. Also making route replies to include a list of node identifiers, initially set to the ancestor list of the responding node will be helpful.

2) Request Propagation: For CBR to work, a single tree must be built and maintained to span all the nodes. It is needed to avoid loops in the propagation of request. This can be achieved by allowing route request packet to jump off the tree at most once.

3) Reply Propagation and Link Activation: It is also needed to avoid the useless propagation of the activation message. This can be achieved by setting the onTree flag that rewrites the header of the route reply message by replacing the identifier of the replying node with its own identifier.

4) Group Leader Election: If multicast activation message is lost the system may end up with a network partition left without a leader. A partition without a leader will never be able to merge. So setting timeout on the reception of activation message will serve the purpose. Group leader can be elected by exploiting the acyclic topology of the tree and electing group leader as the node whose distance from the former leader is minimal.

In order to achieve security against black hole attack, the IDS nodes play an important role. It is necessary to maintain two tables – RQ table for storing
the RREQs of the nodes in its transmission range, and SN table for storing the information of suspicious nodes. Following is the procedure for IDS nodes.

1) IDS node sniffs an RREQ packet and ensures its entry in the RQ table. The RQ table contains an entry of source, destination, source sequence no., broadcasting nodes, expiration time.

2) IDS node sniffs an RREP packet and checks if RREP forwarding node is the destination node. If yes, no processing is required. If not, then (source node, destination node) in RREP are indexed to inquire of the RQ table in following three cases.

Case 1: If there is no corresponding entry in the RQ table, it indicates the RREP forwarding node is not within the transmission range of the IDS that previously broadcasted the RREQ. The algorithm stops without subsequent processing.

Case 2: If there is corresponding entry in the RQ table, and the broadcasting nodes field contains the ID of a RREP forwarding node; it indicates that this is a reasonable reply to RREQ. The algorithm stops without subsequent processing.

Case 3: If there is corresponding entry in the RQ table, and the broadcasting nodes field does not contain the ID of the RREP forwarding node; it indicates this is not a reasonable reply, thus it must inquire about the RQ table by this RREP forwarding node. If this entry exists in the SN table, checks if the status is active, if active, it stops with no further handling. Otherwise, the suspicious value of the entry in the SN table is added with 1, and then checks if this value reaches the threshold. If yes, the status is set active and a Block message is broadcasted. If this entry doesn’t exist in the SN table, a new entry is added in the SN table, and the ID of RREP forwarding node is entered, the suspicious node value is set as 1, and the status is set as inactive.

IV. PERFORMANCE EVALUATION

The performance of the black hole attack on the protocol has been evaluated. Further evaluation of IDS mechanism is yet to be evaluated.

For the implementation of this mechanism, we used NS2 for setting up the simulation environment. The simulation settings are as follows. The network consists of 50 nodes placed randomly within an area of 1000m x 1000 m. Each node moves randomly and has a transmission range of 250m. The random way point model is used as the mobility model. In this model, a node selects a random destination and moves towards that destination at a speed between the predefined maximum and minimum speed. The minimum speed for the simulations is 0 m/s while the maximum speed is 50 m/s. The simulations were carried out with 2, 5, 7 and 9 attackers for different number of receivers. The malicious nodes were selected randomly. The performance metric used is Packet Delivery Ratio, that is, the ratio of the number of data packets delivered to the destinations to the number of data packets generated by the sources. Figure 2 shows the variation of packet delivery ratio (PDR) with mobility when there are no attackers in the presence of 10, 20, 30 and 40 receivers. It is seen that even in the absence of attackers there is a drop in the value of PDR as the mobility increases and an increase in PDR as the numbers of receivers increase.

![Figure 1: PDR under no attackers.](image1)

As the number of attackers increase, the PDR reduces considerably. Figure 3 shows a drop in PDR when the numbers of attackers are increased.

![Figure 2: PDR for 20 receivers.](image2)

V. CONCLUSION

The paper attempts to propose a protocol that can enable Content Based Routing in MANET by organizing the nodes of the network in a single self-repairing tree topology. It also aims to detect and separate black hole nodes by deploying IDS nodes in
MANET. The protocol tolerates the dynamics of the underlying physical network characteristic of MANETs. Also it secures the nodes of MANET from black hole attack. In this paper, the effect of black hole attack on MANETs has been analyzed. The results show that the presence of black hole nodes reduces the packet delivery ratio of the network considerably and affect the performance of the network. The IDS mechanism will help to improve the PDR so that the performance of MAODV improves.

REFERENCES


Whom Should I Display My Actions?  
And, Whose Actions Should I Monitor? In Consciousness Network

B. Deena Divya Nayomi1, S.Masthan Vali2, D.K. Shareef3 & B.Swarun Raj4

1&2Dept. of CSE, SKTRM College of Engineering, Mahaboob Nagar  
3Dept. of CSE, INTELL Engineering College, Anantapur  
4Dept. of CSE, kv subbareddy Engineering College, Kurnool  
E-mail: deena.divya20@gmail.com1, smv1219@gmail.com2, dkshareef19@gmail.com3, rajs.swarun@gmail.com4

Abstract – The concept of awareness has come to play a central role in CSCW research. The coordinative practices of displaying and monitoring have received attention and have led to different venues of research, from computational tool support, such as media spaces and event propagation mechanisms, to ethnographic studies of work. However, these studies have overlooked a different aspect of awareness practices: the identification of the social actors who should be monitored and the actors to whom their actions should be displayed. The focus of this paper is on how social actors answer the following questions: to whom should I display my actions? And, whose actions should I monitor? Ethnographic data from two software development teams are used to answer these questions. In addition, we illustrate how software developers’ work practices are influenced by three different factors: the organizational setting, the age of the project, and the software architecture.

I. INTRODUCTION

Schmidt (2002) discusses some important findings about the concept of awareness recognized by the CSCW community. These findings are based on seminal studies of work practice (Harper, Hughes et al. 1989; Heath and Luff 1992; Heath, Jirotka et al. 1993), and they conceptualize awareness as a range of coordinative practices performed by competent actors to accomplish their work (Heath, Svensson et al. 2002). The nature of these coordinative practices is dual: it involves (i) displaying one’s actions, and (ii) monitoring others’ actions. That is to say, social actors monitor their colleagues’ actions to understand how these actions impact their own work and, while doing their work, social actors display their actions in such a way that others can easily monitor them. The displaying and the monitoring of activities are thus complementary aspects: the displaying of one’s actions is facilitated by the monitoring of the others and vice versa. The practices by which social actors became aware of their colleagues’ work usually have been associated with actors’ achievements—“hidden” results of work arrangements—and not viewed as the result of deliberate, explicit actions (Schmidt 2002). However, this is not the case. In fact, according to Schmidt, social actors deftly choose the degree of obtrusiveness of their actions: “no clear distinction exists between, on the one hand, the coordinative practices of monitoring and displaying, normally referred to under the labels ‘mutual awareness’ or ‘peripheral awareness’, and, on the other hand, the practices of directing attention or interfering for other purposes. In fact, by somehow displaying his or her actions, the actor is always, in some way and to some degree, intending some effect on the activities of colleagues. The distinction is not categorical but merely one of degrees and modes of obtrusiveness.” Despite the undeniable importance of these findings, one aspect has not received enough analytical attention by the CSCW community: the identification of social actors involved in the coordinative practices of awareness, that is, how social actors identify the colleagues who should be monitored and those colleagues to whom their actions should be displayed. We argue that a change of focus is required: instead of focusing on the coordinative practices, one should focus on how social actors answer the following questions: to whom should I display my actions? And, whose actions should I monitor? It is also necessary to understand how the organizational setting facilitates the identification of these two sets of actors. These questions have been looked at from a technological point of view in event notification servers (Lövstrand 1991; Fitzpatrick, Kaplan et al. 2002), usually through subscriptions that allow one to define the notifications to receive.

Empirical studies, however, have not focused on these aspects, partly because the studies of work practice that helped to establish the concept of awareness used the perspectives of ethnomethodology and conversation.
Two different organizations and projects were integrated after the second data collection, datasets from the described next. Details about each team as well as the methods used are grounded theory techniques (Strauss and Corbin 1998). Therefore, we explicitly tried to collect information about this aspect. Data analysis was conducted by using structured interviews (McCacken 1988) for data collection. The role of the software architecture in the work practices was evident during the data collection; how these aspects influence the practices by which they become aware of the actions of their colleagues. We also discuss how organizational settings facilitate or hinder the identification and maintenance of awareness networks. In this regard, this paper briefly illustrates how software developers’ knowledge about the software architecture is used to guarantee a smooth flow of work.

The remainder of this paper is organized as follows. The next section describes the two research sites studied, Alpha and Beta, as well as the methods used to collect and analyze data from these sites. Next, the ethnographic data of the Beta and Alpha teams is presented, and a discussion follows in the subsequent sections. Finally, the last section presents the final comments and future work.

II. RESEARCH SITE AND METHODS

We conducted two qualitative studies at different large software development organizations. The first field study was conducted during summer 2002, and the second one was performed during summer 2003. We adopted observation (Jorgensen 1989) and semi-structured interviews (McCacken 1988) for data collection. The role of the software architecture in the work practices was evident during the data collection; therefore, we explicitly tried to collect information about this aspect. Data analysis was conducted by using grounded theory techniques (Strauss and Corbin 1998). Details about each team as well as the methods used are described next.

III. DATA ANALYSIS

After the second data collection, datasets from the two different organizations and projects were integrated into a software tool for qualitative data analysis, MaxQDA2. After that, the data collected was analyzed by using grounded theory (Strauss and Corbin 1998) with the purpose of identifying a framework to explain the results observed in both field studies. Interviews and field notes were coded to identify categories that were later interconnected with other categories.

IV. THE AWARENESS NETWORK IN THE ALPHA TEAM

The Task Assignment

For accountability purposes, all changes in the Alpha software need to be associated with a problem report. A PR describes the changes in the code, the reason for the changes (bug fixing, enhancement, etc.), and who made the changes, among other pieces of information. An Alpha developer is usually delegated new tasks by being assigned to work with one or more PRs. These PRs are reported by other team members, who are responsible for filling in the field “how to repeat,” which describes the circumstances (data, tools, and their parameters) under which the problem appeared. When software developers report a PR, they also might divide a it into multiple PRs that achieve the same goal. This division aims to facilitate the organization of the changes in the source code, separating PRs that affect the released Alpha tools from those PRs that affect tools or processes not yet released. As mentioned in the previous section, each developer is assigned to one or more processes and tends to specialize in that process. A manager will follow this practice and allocate developers to work on PRs that affect “their” respective processes. However, it is not unusual to find developers working in different processes. In this case, Alpha developers need to identify and contact the process owner to find out whether there is a problem in the process. If there is a problem, developers will start working to find a solution to this problem. Even if the problem is straightforward, before committing their code, Alpha developers need to contact process owners to verify, through a code review (a prescription of the software development process), whether their changes in the process are going to impact the work of these process owners.

V. FINDING OUT WHO TO CONTACT

The need to contact process owners means that the developer working with the PR needs to identify the owner of the process being affected. This is not a problem for most developers, who have been working in the project for a couple of years and already know which developers work on which parts of the source code. In contrast, developers who recently joined the project face a different situation because they lack this
knowledge. To handle this situation, newcomers use information available in the team’s mailing list. The software development process prescribes that software developers should send email to this list before integrating their changes in the shared repository. Developers thus associate the author of the emails describing the changes with the “process” where the changes were occurring: Alpha team members assume that if one developer repeatedly performed check-ins in a specific process, it was very likely that he or she was an expert on that process. Therefore, a developer needing help with that process would know who to contact for help.

**Sending Email**

To conclude the work required to make changes in the Alpha software, developers need to inform their colleagues that they are about to commit their changes to the shared repository. This is done by sending an email to the rest of the team. These emails are necessary due to the lack of modularity of the Alpha software: a change in one particular “process” could impact all other “processes.”

**Using Email**

Emails exchanged among team members are also used by software developers to find out whether they have been engaged in parallel development. Parallel development happens when several developers have the same file checked-out and are simultaneously making changes in this file in their respective workspaces. Note that if a developer, John, is engaged in parallel development with another developer, Mary, and Mary already checked-in her changes in the main branch before John did, John will necessarily have received an email from Mary about her check-in’s. By reading these emails, John will be aware that he is engaged in parallel development with Mary because her email describes, among other things, the files that have been checked-in. In this case, John is required to perform an operation known in the Alpha team as a “back merge.” This operation is supported by the configuration management (CM) tool adopted by the team and is required before a developer can merge his or her code into the main branch.

**VI. THE AWARENESS NETWORK IN THE BETA TEAM**

**The Organizational Context**

As mentioned previously, applications developed in the BSC organization should be designed according to a reference architecture based on layers and APIs, so that components in one layer could request services only to components in the layers immediately below them through the services specified in the APIs. By using this approach, changes in one component could be performed more easily because the impact of these changes is restricted to a predefined set of software components. In addition, changes in the internal details of the component can be performed without affecting this component’s clients. As a consequence of this approach, it is not necessary to broadcast changes to several different software developers, but instead just to a small set of them. That is, by decoupling software components, it is possible to facilitate the coordination of the developers working with these components (Conway 1968; Parnas 1972).

Unfortunately, organizational factors decrease the effectiveness of this approach. For example, the large-scale reuse program adopted by BSC leads Beta developers to interact with developers in different teams who can be located anywhere: in the same building, in different cities, or even in different countries. This is necessary to allow software components to be reused within the organization and to reduce software development costs. However, due to the size and geographical distribution of the organization, this was problematic. During our interviews, we found out that Beta server developers do not know who is consuming the services provided by their components, and Beta client developers do not know who is implementing the component on which they depend. Because of that, developers do not receive important information that affects their work (e.g., important meetings they need to attend).

**Finding Out Who to Contact**

In order to identify who they need to contact, developers adopt different approaches. First, they rely on their personal social networks. Managers also play an important role in this process due to their larger social networks. Beta developers contact them so that these managers can identify the person they want to find.

In one occasion, a client developer “followed” his technical dependency in order to switch teams: his software component had a dependency on a component provided by the server team, who actually had a dependency in a component from the infra-structure team, who depended on an external team’s component. To simplify the communication channels and make sure that the client team would have the component, the manager of the client team decided to “lend” this developer to the external team. By doing this, the manager, to some extent, could guarantee that the services he needed would be implemented. This approach provides another advantage: managers would guarantee the stability of part of their awareness network.
Not everything is hectic in the BSC organization, though. An organizational aspect facilitates the identification of the awareness network: the API review meetings. Within the Beta team, these meetings are scheduled to discuss the APIs being developed by the server team. The following people are invited: API consumers, API producers, and the test team that eventually will test the software component’s functionality through this API. In addition to guaranteeing that the API meets the requirements of the client team and that this team understands how to use it, this meeting also allows software developers to meet. After that, the server team provides APIs to the client team with “dummy implementations” to temporarily reduce communication needs between them, thus allowing independent work. This approach is useful only in some cases, due to the time that passes between API meetings and the actual implementation of the API. In the meantime, changes in developers’ assignments may cause communication problems because developers do not know about each other anymore. In short, changes in assignments change the awareness network, thereby making the work of software developers more difficult to coordinate.

VII. CONCLUSION

The term “awareness” is used to describe a range of practices by which social actors coordinate their work through the display of their actions to their colleagues and the monitoring of actions from their colleagues. Most empirical studies focus on the identification of these coordinative practices and assume settings in which the social actors who display and monitor actions do not change often. This happens because the seminal studies of awareness practices usually adopted perspectives (ethnomethodology or conversation analysis) that focused on actions in a small time frame. Furthermore, these studies focused on settings such as control rooms, newsrooms, and trading rooms, which have characteristics that make necessary for individuals to monitor each other’s conduct on an ongoing basis while engaged in distinct but related activities (Heath, Svensson et al. 2002).

REFERENCES


Dynamic Detection of Design Inconsistency During Software Development Using Daid Approach

Suma V¹, Arfa Baig², Divyashree B J³, N B Sonali⁴ & S Akshaya⁵
¹Research and Industry Incubation Center, ²Dept. of Information Science and Engineering, Dayananda Sagar Institutions, Bangalore, India
E-mail : ¹sumavdsce@gmail.com, ²arfa.61@gmail.com,³divyabj08@gmail.com,⁴sonalinayakank@gmail.com,⁵akshayashankar@gmail.com

Abstract – Evolution of software has lead to the fast growth of technology whose impact can be witnessed in all the domains of scientific and engineering applications. Hence engineering high quality software is one of the core challenges of all IT industries. The software models which are being used for the development of the software products may lead to inconsistencies. Nevertheless, the existence of several methodologies during the development process in order to overcome inconsistencies operates at static mode leading towards expensive nature of rework on those inconsistencies. Therefore, this paper presents a dynamic model which resolves the aforementioned issue by capturing inconsistencies dynamically in an automated mode using Dynamic automated inconsistency detection (DAID) model. The implementation results of DAID capture the design inconsistencies dynamically at the time of their injection points in lieu of inconsistency detection during validation testing. This approach of dynamic design inconsistency detection reduces cost, time and its associated overheads. Further implementation of DAID in an automated mode increases productivity, quality and sustainability in IT industries.

Keywords - Software Quality; Consistency Checking; Software Development Life Cycle; Design Techniques, Analysis of inconsistencies

I. INTRODUCTION

For the persistence of software industry, total customer satisfaction is necessary. Hence emergence of high quality software is necessary. Attributes of high quality software includes development of software product within the scheduled budget, resources, time and more importantly to be flaw free. Hence it is rudimentary for any software generating organization to implement highly structured development process. Since quality does not emerge as an end factor but is realized as a continual process, the entire software development life cycle comprising of requirements analysis phase, design phase, incrimination phase and testing should be individually and cumulatively defect free. Further, most of the IT industries use objects oriented approach to implement their applications. It is worth to note that designing software models enables one to convert the theoretical requirement specifications into implementable code. However, from our adherent interpretation on several projects across several software industries indicate that the design phase is more error prone due to the existence of conventional mode of designing. Currently, there exists several design models to detect design. The current approach of design models uses batch consistency and type based approaches which detects the inconsistencies in the design model. However, these approaches have a major limitation as it is time consuming and use manual annotations. It further leads to accelerated cost with diminished quality and productivity which is a foremost risk for any software company. Hence, inconsistency detection in design models should be carried out to avoid unnecessary rework. However, it is seen that the problem of delayed detection of design inconsistency is due to lack of automated mode of detecting inconsistency. Further, they unearth these inconsistencies during validation testing rather than their detection dynamically at the time of their injection points. Hence the scope of this research is to reduce injection of design inconsistencies dynamically. This paper introduces the development of an automated design checking inconsistency checking technique using Dynamic automated inconsistency detection (DAID) model. The implementation results of DAID model is presented in this paper using a sample ATM case study. Section II of this paper provides background work for the above-stated problem. Section III elucidates the research design and Section IV provides the description of the developed model.
Section V provides implementation of DAID model and results. Section VI summarizes the entire work.

II. LITERATURE SURVEY

Several researches is progressive in detecting the inconsistencies in software models. Authors in [1], state that the consistency is performed using inter-view point rules to detect the inconsistencies [1]. To make the consistency checking easier authors of [2] have used graph representation comprising of class and sequence diagram [2]. However, author in [3] recommends UML design models tend to detect and repair inconsistencies and they further suggest the designer to be aware of design flaws inorder to overcome the design inconsistencies [3]. Authors of [4] has proposed technique to tolerate and manage inconsistencies during software development [4]. Authors in paper [5] provide various options and approaches for fixing inconsistencies [5]. Author in [6] suggests use of various metamodels and consistency rules for fixing inconsistencies during development activity [6]. However, author in [7] has introduced a new technique to automatically decide when to evaluate the consistency rules and works with black box consistency rules [7]. Authors in [8] expresses that regression testing techniques to identify inconsistencies and other flaws existing in the software [8]. Authors in [9] state that major defects are seen at design phase for the applications developed in product based industries. Therefore, authors in [10] have proposed development of dynamic model to reduce design inconsistencies dynamically.

III. RESEARCH DESIGN

Consistency rules are conditions on a model, using which the model is validated. Instantaneous consistency checking thus requires a perceptive of how the model changes. This research proceeds with the identification of design problem in an automated manner by listing out the various inconsistencies related to every part or phase of the designed model. With the knowledge of existing works this research lead us to propose a dynamic and automated technique of detecting the design inconsistencies during software development process. Consequently, the next step was to contrast the existing techniques with respect to cost and time objectives that lead us to major differences stating the efficiency of our approach. From the data analysis, this investigation has directed to introduce an innovative technique through which design inconsistencies are detected in an automated manner without human intervention right at the time of its inception. This is achieved with the implementation of DAID (Dynamic and Automated Inconsistency Detection) model.

IV. DAID MODEL

The architecture of DAID model is shown in Figure 1. This section provides an explanation on the operations of DAID model using a sample ATM example.

Working of DAID model starts with the generation of SRS from the designer followed by the design specifications. Traditionally, the inconsistencies detected are modified in implementation phase which leads to rework. Our model perceives the inconsistencies in the design before the model enters the implementation phase in order to unearth design flaws.

Consider the case study which is an ATM application. DAID Model verifies card using the rule checker. In case an inconsistency is detected (Card is invalid), the rule checker reports the inconsistencies to the designer. The designer in turn, rectifies the model and reverts it back to the DAID model for rechecking. This rechecking mechanism is allowed to happen only for two cycles. This will help us to check the efficiency of the designer based on the design complexity as well as the time constraint. In cases where there are no inconsistencies (such as a card is valid) detected, the above mentioned process is not performed and the design directly enters code construction phase.

![DAID Model Diagram]

**Fig. 1 : DAID Model [10]**

An ATM application has several classes and in this paper a sample class is depicted in figure 2. Accordingly, accounts class can be either of type savings account or current account. There can be inconsistencies observed during the design of this class.
Table 1 depicts a sample list of design inconsistencies that may occur in the sampled ATM application.

<table>
<thead>
<tr>
<th>Inconsistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>The ATM can’t read the card.</td>
</tr>
<tr>
<td>The card has expired.</td>
</tr>
<tr>
<td>The ATM times out waiting for a response.</td>
</tr>
<tr>
<td>The amount is invalid.</td>
</tr>
<tr>
<td>The account is invalid.</td>
</tr>
<tr>
<td>The machine is out of cash or paper.</td>
</tr>
<tr>
<td>The communication lines are down.</td>
</tr>
<tr>
<td>Transaction is rejected of card duplication.</td>
</tr>
</tbody>
</table>

Table 1. Sample inconsistency list of ATM application

Figure 3.1, Figure 3.2, Figure 3.3, Figure 3.4 and Figure 3.5 illustrate the sequence diagrams for various sampled inconsistencies observed in ATM application.

From the Figure 3.1, Figure 3.2, Figure 3.3, Figure 3.4 and Figure 3.5, we can infer that there can be inconsistencies such as rejection of card, invalid amount, out of cash and duplication of card which needs to be addressed at the time of their injection in order to reduce overheads during software development process.
V. RESULTS

Consider the following UML Diagram of an ATM application which is constructed in the ArgoUML software. ArgoUML allows us to construct any UML diagram which can be exported and given as input in SDMetrics.

The aim is to provide a high level comprehensive view of transformation of user requirements into consistent design specifications using a simplified UML model which is constructed in ArgoUML and it can be in the form of class diagram, sequence diagram and also use case diagram.

The completed design is then saved and in order to check for the inconsistencies, this file is exported with the help of XMI (XML metadata interchange) which enables easy interchange between ArgoUML and SDMetrics.

SDMetrics analyzes the structure of the UML models by making use of the design rule checking to automatically detect incomplete, incorrect, redundant or inconsistent design. It finds problems at the design stage, even before they are committed to source code.

Rule Checker:

The rules shown in the following table are cross checked by the rule checker against the UML diagram which is given as input.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttrNameOvr</td>
<td>Naming</td>
<td>Class defines a property of the same name as an inherited attribute</td>
</tr>
<tr>
<td>CyclicInheritance</td>
<td>Inheritance</td>
<td>Class inherits from itself directly/indirectly</td>
</tr>
</tbody>
</table>

Table 2: Rules and their description

Report Generator

Once the model is analyzed with the help of the rules, the list of inconsistencies will be displayed by the report generator which checks whether the given model is consistent.

Once the rule checking for each model element in the UML diagram is completed, the metrics is calculated and is displayed as a table which is shown below in Figure 5:

Fig. 5: Snapshot of list of inconsistencies

Figure 6 and Figure 7 illustrate the cost and time analysis using DAID.
VI. CONCLUSION

Development of quality software is one of the trivial activities of any software industry. However, the current scenario in most of the IT industries focuses on quality measurement at the final stages and not at all the phases. Since, design phase is one of the injections and dwelling point for several design inconsistencies, it is imperative to resolve those flaws without enabling them to propagate. There exists several design quality models which operate in the static mode rather than dynamic detection of design flaws. This paper therefore introduces Dynamic and Automated Inconsistency Detection (DAID) model along with consistency rules so as to facilitate fast, accurate, and dynamic detection of design inconsistencies using the predefined design rules right at the design phase itself. The approach described here works on UML models that can be exported to various formats such as XMI (XML Meta data interchange) and their consistency can be checked through software such as SDMetrics. Using the DAID model, the efficiency of the designer can be measured. Further, the complexity of the code and incomplete requirements specification can be tested. This approach further enhances the productivity and the quality of the product in the industry with reduction in cost and time for rework of defects.

REFERENCES


Secure Message Communication for Battlefield Soldiers Using Android

Manjunath Shirageri & Pradeep N
Department of Computer Science, BIET, Davangere, Karnataka, India
E-mail : manju.shirageri@gmail.com, nmnpradeep@yahoo.com

Abstract – Message Service is getting more popular now-a-days. Message (SMS) was first used in December 1992, when Neil Papworth, a 22-year-old test engineer used a personal computer to send the text message "Merry Christmas" via the Vodafone GSM network to the phone of Richard Jarvis in the UK. It will play a very important role in the future business areas of mobile commerce (M-Commerce). SMS’s security has become a major concern for business organizations and customers.

Many people send delicate information and conduct private conversations via text with little protection from third parties who might intercept the message (SMS) or the storage of their information in phone company records but this is not case for soldiers. The message (SMS) communication between soldiers in order to fill this void and offer soldiers a more securely private means of textual communication, Secure Message Communication for Battlefield Soldiers Using Android is developed.

We used ECC cryptosystem for encryption and decryption of message (SMS). Text Encryption will be a third-party application capable of running on any Android system. It will allow users to send and receive encrypted text messages using our application. About key exchange we used Diffie Hellman key exchange mechanism it’ll allow automatically exchange key between soldiers and start a secure session. In this way, we hope to provide a safe and secure means of transferring private messages between any two Android phones and it’ll also provide identifying end user as a valid user or not.

Keywords - Android, Decryption, Encryption, ECC, SMS, Text Secure.

I. INTRODUCTION

Messaging (SMS) is getting more popular now-a-days. It will play a very important role in the mobile messa mobile commerce [3] (M-Commerce). Up to now many business organizations use SMS for their business purposes. SMS’s security has become a major concern for business organizations and customers. There is a need for an end to end SMS Encryption in order to provide a secure medium for communication. Security is main concern for any business company such as banks who will provide these mobile banking services. Currently there is no such scheme that provides complete SMSs security.

The mobile messaging market is growing rapidly and is a very profitable business for mobile operators. It can be seen from figure1 that the growth rate of SMS in worldwide during 2000 – 2015F (F stands for forecast) in billion.

SMS has a variety of advantages and disadvantages for M-Commerce purpose [3]. The advantages are easy to use, common messaging tool among consumers, works across all wireless operators, affordable for mobile users, no specific software required to installation, allows banks and financial institutions to provide real-time information to consumers & employees, stored messages can be accessed without a network connection.

Very few disadvantages are text data and limited up to 140-160 characters per message, does not offer a secure environment for confidential data during transmission and there is no standard
procedure to certify the SMS sender. Presently researchers proposed some security concepts regarding SMS security. Most of the proposals are software frames to be installed on mobile device and/or on the SIM cards to implement security [4].

Two are the major security vulnerabilities affecting SMS [1] based communication: the lack of confidentiality during the transmission of a message and the absence of a standard way to certify the identity of the user (or at least his phone number) who sent the message.

This project regarding to exchange message in secure manner at peer level. It has a software framework to enable user to transfer message in secure manner using ECC [5] and security parameters for transmitting secure message to achieve better cost and efficiency of the operation.

This represented in external architecture of project as shown in figure 2.

The remaining part of the paper is organized as follows: Section II gives a view about secure messaging, Section III describes the System Architecture, Section IV describes the Implementation details, and finally Section V gives the conclusion and future work.

II. SECURE MESSAGING

Project is based on non-server [12] architecture mobile communications, security solutions are implementable for individuals due to its independency from the mobile phone network operator or service provider. Thus, the user does not need to make any agreement with the mobile phone network operator or service provider and use of ECC cryptosystem [4] through a non-server[12] based architecture makes better choice to easily experiment. As a result, all the cryptographic operations are achieved on the user’s mobile phone. Terms of overhead cost of communication is less than server architecture system, due to discard in the communication between the user and the server.

Secure messaging will be a third-party application capable of running on any Android system. It will allow users to send and receive encrypted text messages using the standard SMS [10] text messaging system and will only sends encrypted data over that system. Without the secure SMS program and an appropriate key/password, any intercepted or stored messages will appear unreadable. In this way, we hope to provide a safe and secure means of transferring private messages between any two Android phones.

III. SYSTEM ARCHITECTURE

The SMS system is a service provided by mobile network company. Our main motto is implementing a security[4] at application level this process contains 4 fundamental elements sender, receiver, encryptor and decryptor which are shown in figure 3.

**Sender:** The sender using “Secure Message Communication for Battlefield Soldiers Using Android” application allowed to enter passphrase for generating a local key value. After generating key value sender can establish a secure session with other device. This secure session indicates an encrypted SMS transfer between sender and receiver.

**Receiver:** As soon as message arrived from sender to receiver, receiver is intimated by a notification message indicating that key exchange and processing of key is completed and secure session can start now. Both at sender and receiver encryption (while sending) and decryption (while receiving) occurs automatically.

**Encryptor and Decryptor:** This takes a text message entered by user and key which is exchanged between two parties before session begins. Encryptor module automatically encrypts SMS and sends that to receiver. Sender also no where aware about cipher pattern. Where as decryptor takes a cipher text which is received from SMS System and takes key which is exchanged before secure session established. As soon as cipher SMS is received from sender, the decryptor module converts that into plane text and displays in device screen.

**SMS System:** This is built in mobile network where it performs a store forwarding of message to or from end users.

**Shared key:** This key is generated at both end automatically by using a local key pairs generated in
communicating parties. This generated local key pairs are exchanged by using Diffi Hellman key exchange algorithm.

Fig. 3: System architecture

IV. IMPLEMENTATION

Secure message application developed using Android JAVA and which uses crypto packages from java library. The implementation includes dividing complete project into four modules they are sender, receiver, encryption and decryption modules and GUI for sender is such that sender is allowed to enter receiver mobile number, raw key and message which is to be transmitted while at receiver end, receiver is allowed to enter same raw key to decrypt and read. The coding keeps users away from the internal key generation and encryption/decryption part this makes project simple and efficient.

V. RESULTS & SNAPSHOT

Transferring a encrypted messages in a secure session with a less delay by automatic encrypting and decrypting the message before transmission begins and also identifying a man in middle attack.

The following snapshots show the project output which is developed for DRDO.

Fig 1 indicates installation of project.

Fig 2 acceptation of rules, if refused then project will not get install.

Fig 3 shows the dialog which accepts passphrase value from Soldiers to generate local encryption key.

Fig 4 shows this toast message showing a generation of encryption key pair.

Fig 5 shows dialog showing establishment of secure session with particular soldier.

Fig 6 shows automatic key exchange message between two parties. This indicates a secure session is established.

Fig 7 shows dialog showing identification of session. This feature helps to identify man in middle attack.

Fig 8 shows menu options i.e verify recipients identity, verify secure session and aborting session.
REFERENCES


+++
Portfolio Selection in Multipath Routing for Traffic Allocation

M. Karunakar Reddy & S. Nageswara Rao
Madanapalle Institute of Technology and Science, Madanapalle

Abstract – Multiple-path source routing protocols allow a data source node to distribute the total traffic among available paths. In this article, we consider the problem of jamming-aware source routing in which the source node performs traffic allocation based on empirical jamming statistics at individual network nodes. We formulate this traffic allocation as a lossy network flow optimization problem using portfolio selection theory from financial statistics. We show that in multi-source networks, this centralized optimization problem can be solved using a distributed algorithm based on decomposition in network utility maximization (NUM). We demonstrate the network’s ability to estimate the impact of jamming and incorporate these estimates into the traffic allocation problem. Finally, we simulate the achievable throughput using our proposed traffic allocation method in several scenarios.

Keywords - Jamming, Multiple path routing, Portfolio selection theory, Optimization, Network utility maximization

I. INTRODUCTION

Jamming point-to-point transmissions in a wireless mesh network [1] or underwater acoustic network [2] can have debilitating effects on data transport through the network. The effects of jamming at the physical layer resonate through the protocol stack, providing an effective denial-of-service (DoS) attack [3] on end-to-end data communication. The simplest methods to defend a network against jamming attacks comprise physical layer solutions such as spread-spectrum or beamforming, forcing the jammers to expend a greater resource to reach the same goal. However, recent work has demonstrated that intelligent jammers can incorporate crosslayer protocol information into jamming attacks, reducing resource expenditure by several orders of magnitude by targeting certain link layer and MAC implementations [4]–[6] as well as link layer error detection and correction protocols. The majority of ant jamming techniques make use of diversity. For example, ant jamming protocols may employ multiple frequency bands, different MAC channels, or multiple routing paths. Such diversity techniques help to curb the effects of the jamming attack by requiring the jammer to act on multiple resources simultaneously. In this paper, we consider the ant jamming diversity based on the use of multiple routing paths. Using multiple-path variants of source routing protocols such as Dynamic Source Routing (DSR) or Ad Hoc On-Demand Distance Vector (AODV), for example the MP-DSR protocol, each source node can request several routing paths to the destination node for concurrent use. To make effective use of this routing diversity, however, each source node must be able to make an intelligent allocation of traffic across the available paths while considering the potential effect of jamming on the resulting data throughput. In order to characterize the effect of jamming on throughput, each source must collect information on the impact of the jamming attack in various parts of the network. However, the extent of jamming at each network node depends on a number of unknown parameters, including the strategy used by the individual jammers and the relative location of the jammers with respect to each transmitter–receiver pair. Hence, the impact of jamming is probabilistic from the perspective of the network, and the characterization of the jamming impact is further complicated by the fact that the jammers’ strategies may be dynamic and the jammers themselves may be mobile. In order to capture the nondeterministic and dynamic effects of the jamming attack, we model the packet error rate at each network node as a random process. At a given time, the randomness in the packet error rate is due to the uncertainty in the jamming parameters, while the time variability in the packet error rate is due to the jamming dynamics and mobility. Hence, more sophisticated anti-jamming methods and defensive measures must be incorporated into higher-layer protocols, for example channel surfing [8] or routing around jammed regions of the network [6]. Contributions to this problem are as follows:

• We formulate the problem of allocating traffic across multiple routing paths in the presence of jamming as a lossy network flow optimization problem.
• We map the optimization problem to that of asset allocation using portfolio selection theory.
We formulate the centralized traffic allocation problem for multiple source nodes as a convex optimization problem.

- We show that the multi-source multiple-path optimal traffic allocation can be computed at the source nodes using a distributed algorithm based on network utility maximization (NUM).
- We propose methods which allow individual network nodes to locally characterize the jamming impact and aggregate this information for the source nodes.
- We demonstrate that the use of portfolio selection theory allows the data sources to balance the expected data throughput with the uncertainty in achievable traffic rates.

II. BACKGROUND

2.1 Characterizing The Impact Of Jamming

In this Module, the network nodes to estimate and characterize the impact of jamming and for a source node to incorporate these estimates into its traffic allocation. In order for a source node to incorporate the jamming impact in the traffic allocation problem, the effect of jamming on transmissions over each link must be estimated. However, to capture the jammer mobility and the dynamic effects of the jamming attack, the local estimates need to be continually updated.

2.2 Effect of Jammer Mobility on Network

The capacity indicating the link maximum number of packets per second (pkt/s) eg: 200 pkt/s which can be transported over the wireless link. Whenever the source is generating data at a rate of 300 pkt/s to be transmitted at the time jamming to be occurring. Then the throughput rate to be less. If the source node becomes aware of this effect the allocation of traffic can be changed to 150 pkt/s on each of paths thus recovers the jamming path.

Fig. 1: An example network that illustrates a single-source network with three routing paths.

2.3 Estimating End-to-End Packet Success Rates

The packet success rate estimates for the links in a routing path, the source needs to estimate the effective end-to-end packet success rate to determine the optimal traffic allocation. Assuming the total time required to transport packets from each source to the corresponding destination is negligible compared to the update relay period.

2.4 Optimal Jamming-Aware Traffic Allocation

An optimization framework for jamming-aware traffic allocation to multiple routing paths for each source node. We develop a set of constraints imposed on traffic allocation solutions and then formulate a utility function for optimal traffic allocation by mapping the problem to that of portfolio selection in finance.

III. CASE STUDY

3.1 Routing

Routing is the process of selecting paths in a network along which to send network traffic. Routing is performed for many kinds of networks, including the telephone network (Circuit switching), electronic data networks (such as the Internet), and transportation networks. This article is concerned primarily with routing in electronic data networks using packet switching technology.

3.2 Routing protocol:

A routing protocol is a protocol that specifies how routers communicate with each other, disseminating information that enables them to select routes between any two nodes on a computer network, the choice of the route being done by routing algorithms. Each router has a priori knowledge only of networks attached to it directly. A routing protocol shares this information first among immediate
neighbors, and then throughout the network. This way, routers gain knowledge of the topology of the network.

3.3 Multiple Path Routing

Multipath routing is the routing technique of using multiple alternative paths through a network, which can yield a variety of benefits such as fault tolerance, increased bandwidth, or improved security. The multiple paths computed might be overlapped, edge-disjointed or node-disjointed with each other.

IV. CONCLUSION AND FUTURE WORK

We studied the problem of traffic allocation in multiple-path routing algorithms in the presence of jammers whose effect can only be characterized statistically. We have presented methods for each network node to probabilistically characterize the local impact of a dynamic jamming attack and for data sources to incorporate this information into the routing algorithm. We formulated multiple-path traffic allocation in multi-source networks as a lossy network flow optimization problem using an objective function based on portfolio selection theory from finance. We showed that this centralized optimization problem can be solved using a distributed algorithm based on decomposition in network utility maximization (NUM). We presented simulation results to illustrate the impact of jamming dynamics and mobility on network throughput and to demonstrate the efficacy of our traffic allocation algorithm. We have thus shown that multiple path source routing algorithms can optimize the throughput performance by effectively incorporating the empirical jamming impact into the allocation of traffic to the set of paths.

We propose a scheme based on multiple routing paths. The wireless network of interest can be represented by a directed graph. The solution is when a source node S want to send data to a target node T, it finds all the paths to route the packet from S to T. The traffic to be sent from S to T is split and sent across multiple paths. Say there is 100 packets to be sent, how many packets to sent in each routing path is to be decided. The logic of how to split the traffic across multiple paths takes into consideration the expected jamming in each path. The algorithm to solve is called as Optimal Jamming aware traffic allocation.

REFERENCES

A Review on Emerging Trends in Wireless Networks

R. Sujitha Rani & YSSR Murthy
Department of Computer Science & Engineering, Shri Vishnu Engineering College for Women, Vishnupur, Bhimavaram, West Godavari, AP, India
E-mail : suji.rimmanapudi@gmail.com, yssrmoorthy@gmail.com

Abstract – In this rapidly increasing world of technology, there is an enormous growth in the field of wireless technologies. The demand for wireless communication systems of increasing sophistication and ubiquity has led to the need for a better understanding of fundamental issues in communication theory and electromagnetic systems and their implications for the design of highly-capable wireless systems. A great deal of research is taking place in various aspects of mobile and wireless communications. This paper is an attempt to present a few of these technologies which have a great industrial impact as it is not possible to cover all the emerging trends. The paper presents the main characteristics and design issues and provides a description of wireless communication networks like Powerline Communication (PLC), Ultra Wideband (UWB), Free Space Optics (FSO), Mobile Ad Hoc Networks (MANETs), wireless sensor networks (WSNs), and Flash OFDM (Orthogonal Frequency Division Multiplexing).

Keywords - PLC, UWB, FSO, MANETs, WSNs, OFDM;

I. INTRODUCTION

There is a substantial uncertainty about the future generation of various wireless networks and the evolution of mobile networks. There are many remarkable changes taking place in wireless networks. It is obvious that Wireless Networks is witnessed as an unavoidable threat for other and older kind of networking technologies. Wireless networks provide an inexpensive and easy way to share a single Internet connection among several computers. With the help of wireless networks telecommunication networks and enterprise business installations avoid the costly process of introducing cables into a building, or as a connection between various equipment locations. Wireless communication networks are generally implemented and administered using a transmission system. This implementation takes place at the physical level (layer) of the OSI model network structure. Wireless networks are defined on the bases of their size (that is the number of machines), their range and the speed of data transfer. There are several types of networks available such as Wireless PAN, LAN, MAN, WAN, Mobile Devices Networks and Global Systems for Mobile Communications.

The development of wireless networks is still in progress as the usage is rapidly growing. Personal communications are made easy with the advent of cell phones where radio satellites are used for networking between continents. Whether small or big, businesses uses wireless networks for fast data sharing with economical means. Sometimes compatibility issues with new devices might arise in these extremely vulnerable networks but the technology has made the uploading and the downloading of huge data with least maintenance cost.

In this paper, a conceptual framework for the analysis of emerging wireless technologies is constructed, integrating the viewpoints of end-user, technology, value network, and public policy.

II. POWER LINE COMMUNICATION NETWORK

Power line communication (PLC) is a system for carrying data on a conductor that is also used for electric power transmission. It is also known as power line carrier, power line digital subscriber line (PDSL), mains communication, power line telecom (PLT), power line networking (PLN), and broadband over power lines (BPL). A wide range of power line communication technologies are needed for different applications, ranging from home automation to internet access.

Most PLC technologies limit themselves to one set of wires (such as premises wiring within a single building), but some can cross between two levels (for example, both the distribution network and premises wiring). Typically transformers prevent propagating the signal, which requires multiple technologies to form very large networks. Various data rates and frequencies are used in different situations. It is a critical energy control networking technology for Smart Grid, Smart...
Cities, and Smart Buildings applications. PLC has many advantages over other wired and wireless communication methods. However, PLC technology must overcome some significant technical challenges to be practical. Echelon’s experience and unique innovations allow us to offer the world’s most proven, open standard, and multi-application PLC technology.

PLC is appealing because there is no need to run additional wires to powered devices. PLC can also work where radio frequency (RF) cannot. For example smart meters in the basement of a building are unlikely to be able to use RF to communicate with the neighborhood data concentrator. PLC communication on the other hand can traverse the power wires to reach the data concentrator. For these reasons most utilities around the world (with some geographical exceptions in North America and Australia) have chosen PLC for their smart grid projects, and most cities have chosen PLC for their smart street light projects.

III. ULTRA WIDE BAND

Wireless USB replaces traditional USB cables and PC interfaces with a wireless connection based on UWB. The competing UWB-based CableFree USB and Certified Wireless USB (WUSB) standards operate at speeds between 110 Mbps and 480 Mbps depending on distance.

Ultra-wideband (also known as UWB, ultra-wide band, and ultraband) is a radio technology pioneered by Robert A. Scholtz and others which may be used at a very low energy level for short-range, high-bandwidth communications using a large portion of the radio spectrum. UWB has traditional applications in non-cooperative radar imaging. Most recent applications target sensor data collection, precision locating and tracking applications. One way to share wireless high-definition video across a home network is via UWB connections, as an alternative to Wi-Fi. WB’s higher bandwidth links can better handle large volumes of content. Several industry standards for wireless video streaming compete with UWB including Wireless HD (WiHD) and Wireless High Definition Interface (WHDI).

Because its radios require low power to operate, UWB technology can work well in Bluetooth devices. While Bluetooth 1.0 and 2.0 did not utilize UWB, newer high-speed versions of Bluetooth may.

The limited range of UWB signals preclude it being used for direct connections to internet hotspots. However, some industry efforts existed to enable cell phones with UWB for peer-to-peer applications.

IV. FREE SPACE OPTICS

Free-space optical communication (FSO) is an optical communication technology that uses light propagating in free space to transmit data for telecommunications or computer networking. "Free space" means air, outer space, vacuum, or something similar. This contrasts with using solids such as optical fiber cable or an optical transmission line. The technology is useful where the physical connections are impractical due to high costs or other considerations.

The theory of FSO is essentially the same as that for fiber optic transmission. The difference is that the energy beam is collimated and sent through clear air or space from the source to the destination, rather than guided through an optical fiber. If the energy source does not produce a sufficiently parallel beam to travel the required distance, collimation can be done with lenses. At the source, the visible or IR energy is modulated with the data to be transmitted. At the destination, the beam is intercepted by a photo-detector, the data is extracted from the visible or IR beam (demodulated), and the resulting signal is amplified and sent to the hardware.

FSO systems can function over distances of several kilometers. As long as there is a clear line of sight
between the source and the destination, communication is theoretically possible. Even if there is no direct line of sight, strategically positioned mirrors can be used to reflect the energy. The beams can pass through glass windows with little or no attenuation (as long as the windows are kept clean!). Although FSO systems can be a good solution for some broadband networking needs, there are limitations. Most significant is the fact that rain, dust, snow, fog, or smog can block the transmission path and shut down the network.

V. WIRELESS SENSOR NETWORKS

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, humidity, motion or pollutants and to cooperatively pass their data through the network to a main location. The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a micro controller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created.

Since the start of the third Millennium, wireless sensor networks (WSNs) generated an increasing interest from industrial and research perspectives. A WSN can be generally described as a network of nodes that cooperatively sense and may control the environment enabling interaction between persons or computers and the surrounding environment. On one hand, WSNs enable new applications and thus new possible markets, on the other hand, the design is affected by several constraints that call for new paradigms. In fact, the activity of sensing, processing, and communication under limited amount of energy, ignites a cross-layer design approach typically requiring the joint consideration of distributed signal/data processing, medium access control, and communication protocols.

The main features of WSNs, as could be deduced by the general description given in the previous sections, are: scalability with respect to the number of nodes in the network, self-organization, self-healing, energy efficiency, a sufficient degree of connectivity among nodes, low-complexity, low cost and size of nodes. Those protocol architectures and technical solutions providing such features can be considered as a potential framework for the creation of these networks, but, unfortunately, the definition of such a protocol architecture and technical solution is not simple, and the research still needs to work on it.

VI. MOBILE ADHOC NETWORKS

A mobile ad-hoc network consists of mobile hosts equipped with wireless communication devices. The transmission of a mobile host is received by all hosts within its transmission range due to the broadcast nature of wireless communication and omni-directional antennae. If two wireless hosts are out of their transmission ranges in the ad hoc networks, other mobile hosts located between them can forward their messages, which effectively builds connected networks among the mobile hosts in the deployed area. Due to the mobility of wireless hosts, each host needs to be equipped with the capability of an autonomous system, or a routing function without any statically established infrastructure or centralized administration. The mobile hosts can move arbitrarily and can be turned on or off without notifying other hosts. The mobility and autonomy introduces a dynamic topology of the networks not only because end-hosts are transient but also because intermediate hosts on a communication path are transient.

Characteristics:
- Operating without a central coordinator
- Multi-hop radio relaying
- Frequent link breakage due to mobile nodes
- Constraint resources (bandwidth, computing power, battery lifetime, etc.) Instant deployment

MANets – Mobile Ad-hoc Networks. It’s the classical ad hoc network that came from the military sector. These networks were developed to connect planes, tanks, troops at the battlefield. They are completely self-organizing. The other possible areas where they could be applied are disaster recovery, car-to-car communication, home networking. This class of ad hoc networks can be seen as the forefather of the other classes.

MANets have no fixed routers, every node could be router. All nodes are capable of movement and can be connected dynamically in arbitrary manner. The responsibilities for organizing and controlling the network are distributed among the terminals themselves. The entire network is mobile, and the individual terminals are allowed to move freely. In this type of networks, some pairs of terminals may not be able to communicate directly with each other and have to rely on some terminals so that the messages are delivered to
their destinations. Such networks are often referred to as multi-hop or store-and-forward networks. The nodes of these networks function as routers, which discover and maintain routes to other nodes in the networks. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices.

Mobile Ad-hoc Networks are supposed to be used for disaster recovery, battlefield communications, and rescue operations when the wired network is not available. It can provide a feasible means for ground communications and information access.

**Fundamental Challenges of Wireless Ad-hoc Networks:**

Since Wireless Ad-hoc Networks are inherently different from the well-known wired networks, it is an absolutely new architecture. Thus some challenges raise from the two key aspects: self-organization and wireless transport of information. First of all, since the nodes in a Wireless Ad-hoc Network are free to move arbitrarily at any time. So the networks topology of MANET may change randomly and rapidly at unpredictable times. This makes routing difficult because the topology is constantly changing and nodes cannot be assumed to have persistent data storage. In the worst case, we do not even know whether the node will still remain next minute, because the node will leave the network at any minute.

Bandwidth constrained is also a big challenge. Wireless links have significantly lower capacity than their hardwired counterparts. Also, due to multiple access, fading, noise, and interference conditions etc. the wireless links have low throughput.

Energy constrained operation. Some or all of the nodes in a MANET may rely on batteries. In this scenario, the most important system design criteria for optimization may be energy conservation.

Limited physical security: Mobile networks are generally more prone to physical security threats than are fixed cable networks. There is increased possibility of eavesdropping, spoofing and denial-of-service attacks in these networks.

A. Types of MANET

Vehicular Ad-hoc Networks (VANETs) are used for communication among vehicles and between vehicles and roadside equipment. Intelligent vehicular (InVANETs) are a kind of artificial intelligence that helps vehicles to behave in an intelligent manner during vehicle-to-vehicle collisions, accidents, drunken driving etc. Internet Based Mobile Ad-hoc Networks (iMANET) are ad-hoc network ks that link mobile nodes and fixed Internet-gateway nodes. In such type of networks normal ad hoc routing algorithms don't apply directly.

**VII. FLASH OFDM (Orthogonal frequency division multiplexing)**

FLASH-OFDM (FLASH (Fast Low-latency Access with Seamless Handoff) OFDM (Orthogonal Frequency Division Multiplexing)), is an innovative air interface technology designed for the delivery of advanced Internet services in the mobile environment. As its name suggests, the technology is based on the OFDM air link, a wireless access method that combines the attributes of its two predecessors — TDMA and CDMA — to address the unique demands posed by mobile users of broadband data and packetized voice applications.

A. The Concept of OFDM:

An OFDM signal consists of a number of closely spaced modulated carriers. When modulation of any form- voice, data etc is applied to a carrier, then sidebands spread out either side. It is necessary for a receiver to be able to receive the whole signal to be able to successfully demodulate the data. As a result when signals are transmitted close to one another they must be spaced so that the receiver can separate them using a filter and there must be a guard band between them. This is not the case with OFDM. Although the sidebands from each carrier overlap, they can still be received without the interference that might be expected because they are orthogonal to each another. This is achieved by having the carrier spacing equal to the reciprocal of the symbol period.
To see how OFDM works, it is necessary to look at the receiver. This acts as a bank of demodulators, translating each carrier down to DC. The resulting signal is integrated over the symbol period to regenerate the data from that carrier. The same demodulator also demodulates the other carriers. As the carrier spacing equal to the reciprocal of the symbol period means that they will have a whole number of cycles in the symbol period and their contribution will sum to zero - in other words there is no interference contribution.

VIII. CONCLUSION

This paper is an attempt to give a brief description of emerging wireless technologies. The analysis of a wide range of wireless networks is given. All these technologies play a very significant role in the industry and compete with one another with respect to their characteristics such as data rates, distance covered, user mobility and frequency used. It is obvious that these technologies will undergo many changes and further research in the arena of wireless technologies is both imminent and challenging.

REFERENCES


[8] Electronic Publication: Digital Object Identifiers (DOIs):


Abstract – The Altera Cyclone family of FPGA provides the ability to perform run time reconfiguration which is known as Dynamic Reconfiguration. This paper concentrates on how to take a complex System-on-Chip design into three different components: fixed hardware, reconfigurable hardware and software, each handled by dedicated sub flow. The method to dynamically reconfigure the pll and application used to see variations in frame rate using LCD Display Controller with FPGA is presented. This flow can be considered a part of a general methodology that can be exploited for the implementation on a complex System on Programmable Chip. An example of application it consists utilizes the reconfigurable clocks generated by PLL and video processing in the reconfigurable module. The architecture of the proposed application will presented in this paper were prototyped using a Cyclone III Starter Board, which is based on Nios II Embedded Evaluation Kit, Cyclone III Edition.

Keywords - FPGA, Dynamic Reconfiguration, VGA Controller, Video Processing, SOPC.

I. INTRODUCTION

The process of digital systems design has changed dramatically since the introduction of Field Programmable Gate Arrays. The use of FPGAs has been classified broadly into three main categories: rapid prototyping, system implementation, and dynamically reconfigurable subsystems.

Modern Systems on Chip (SoCs) are subject to rapid changes concerning their functionality and other requirements. In order for them to react more flexibly to environmental changes or new tasks, a simple method must be found to allow such systems to adapt to their surroundings. One possible concept for this is based on the assumption that only those parts of the system (system components) which affected by the new tasks or environment must be updated. Following this idea a little further, one can imagine that certain system components could be replaced during run-time, while the remaining, unaffected parts of the system remain fully operational. A developer could use one chip for different tasks and switch between them during run-time. Thus the so called Dynamic Partial Reconfiguration (DPR) leads efficiency of the application development.

A solution for an efficient and reliable reconfigurable clock generator is presented in this paper. The paper begins by providing an overview of PLL in Section II. Section III presents an introduction to dynamically reconfigurable logic and explains the motivation for using it. It reviews the application of dynamically reconfigurable logic and explains how the present state of FPGA technology is impeding the full investigation of more complex applications of dynamic reconfiguration. It also highlights briefly the need for more consistent terminology when describing dynamically reconfigurable systems. Section IV describes a pll reconfiguration scenario in cyclone III fpga and implementation of it. Section V gives the details of the application to see variations in frame rate with LCD display controller using the dynamically reconfigurable PLLs with existing FPGA design tools. Section six concludes the paper.

II. PLL OVERVIEW

A phase-locked loop (PLL) is a closed-loop frequency-control system based on the phase difference between the input clock signal and the feedback clock signal of a controlled oscillator. Figure 1 shows a simplified block diagram of the major components in a PLL. The main blocks of the PLL are the phase frequency detector (PFD), charge pump, loop filter, voltage controlled oscillator (VCO), and counters, such as a feedback counter (M), a pre-scale counter (N), and post-scale counters(C).
The output frequency of the PLL is equal to the VCO frequency ($F_{VCO}$) divided by the post-scale counter (C).

In the form of equations:
- $F_{REF} = F_{IN} / N$
- $F_{VCO} = F_{REF} \times M = F_{IN} \times M / N$
- $F_{OUT} = F_{VCO} / C = (F_{REF} \times M) / (F_{IN} \times M) / (N \times C)$

III. DYNAMIC RECONFIGURATION

Dynamic reconfiguration offers important benefits for the implementation of reconfigurable systems. Dynamically reconfigurable FPGAs offer the fastest possible way to change an active FPGA circuit since only those parts that need to be reconfigured are interrupted. This results in faster overall system operation. The smaller configuration bitstreams also require less external memory for storage. The profile of the circuitry that is active on the array may be adjusted dynamically to match the requirements of the algorithm being executed thus optimising the ratio of active circuitry to available silicon and potentially reducing the overall number of components required by a design. Moreover, the opportunities for deploying dynamic reconfiguration are increasing as the gate counts of individual FPGAs continues to improve. As larger FPGAs become available, the complexity of the systems that can be integrated into a single FPGA increases. Consequently, the probability that the execution of subsets of the logic will be mutually time-exclusive is also likely to increase. The benefits of faster reconfiguration, and hence faster overall system speed, and reduced component count thus become more apparent with increasing part capacity[3].

IV. PLL RECONFIGURATION SCENARIO IN CYCLONE III FPGA

The PLL reconfiguration feature can dynamically adjust the PLL parameters to lock to a very wide spectrum of input clock frequencies. This is particularly useful in display applications where clock rates vary based on the system and the refresh rate chosen. In this case, Altera-provided IP is used to control the
reconfiguration and automatically adjust the PLL parameters to lock onto the given input frequency[5].

A. Implementing PLL Reconfiguration In Cyclone III Devices

PLLs use several divide counters and different voltage-controlled oscillator taps to perform frequency synthesis and phase shifts. In Cyclone III PLLs, you can reconfigure the counter settings and dynamically shift the phase of the PLL output clock. You can also change the charge pump and loop filter components, which dynamically affect the PLL bandwidth. You can use these PLL components to update the clock frequency, PLL bandwidth, and phase shift in real time, without reconfiguring the entire FPGA.

Applications that operate at multiple frequencies can benefit from PLL reconfiguration in real time. PLL reconfiguration is also beneficial in prototyping environments, allowing you to sweep PLL output frequencies and adjust the clock output phase at any stage of the design. For example, a system generating test patterns is required to generate and transmit patterns at 50 or 100 MHz, depending on the device under test. Reconfiguring the PLL components in real time for this example allows you to switch between the two output frequencies within a few microseconds.

The following PLL components are reconfigurable in real time:

- Pre-scale counter (N)
- Feedback counter (M)
- Post-scale output counters (C0-C4)
- Dynamically adjust the charge pump current (ICP) and loop filter components (R, C) to facilitate on-the-fly reconfiguration of the PLL bandwidth

Figure 4 shows how PLL counter settings can be adjusted dynamically by shifting their new settings into a serial shift-register chain or scan chain. Serial data is fed into the scan chain via the scandata port and the shift registers are clocked by the scanclk port. Serial data is shifted through the scan channel as long as the scanclkena signal stays asserted. After the last bit of data is clocked, asserting the reconfiguration state machine signal, configupdate, for at least one scanclk cycle causes the PLL configuration bits to be synchronously updated with the data in the scan registers. The scan chain can also be initialized or changed using a memory initialization file in the hexadecimal (.hex) file or memory initialization file (.mif) format [7].

V. LCD APPLICATION OF DYNAMICALLY RECONFIGURABLE LOGIC

Video systems almost always include an embedded processor and a memory subsystem to manage the video frames in the external memory. The SOPC Builder system tool provided by Altera greatly simplifies embedded system design. This tool includes a library of elements such as soft core processors (Nios® II), interfaces, memory, bridge, and IP cores. It also features a connectivity GUI and generator to automatically wire up arbitrated and streaming bus systems.

Fig. 4 PLL Reconfiguration Scan Chain

A. Nios II Embedded Design Suite

The Nios® II Embedded Design Suite (EDS) is a collection of components and tools used to develop embedded software for the Nios II processor, including Nios II Software Build Tools for Eclipse based on the familiar Eclipse development environment. The Nios II IDE is also included to provide support for legacy designs.

Fig. 5 SOPC-Based Design Flow for Video Datapath
Nios II is an embedded processor system that can be configured by the user, meaning the actual hardware of the processor is easily customized for a particular application through the SOPC Builder feature of the Quartus II and SOPC Builder FPGA design software.

Cyclone III family offers a wide range of capability, including density, on-board memory, and I/O, that designers can make use of in order to design the right application for their end product.

B. **VGA Controller**

The VGA controller core displays images by creating the timing signals required by VGA compatible monitors attached to the VGA port on the DE-series board, or the Terasic LCD screen with touch panel. Video is produced by displaying frames (or images) in rapid succession. In a typical video, frames are displayed between 30 and 120 times per second. A frame is a two-dimensional array of pixels as depicted in Figure 6.

![Fig. 6 Video frame’s screen layout](image)

The resolution of a frame is defined as the number of pixels in the x and y axes. An example resolution is 640x480, which has 640 pixels across the x axis and 480 pixels down the y axis, as shown in Figure 7. Therefore, each pixel location in a frame can be identified by an (x, y) coordinate, with (0, 0) being in the top-left corner [8].

![Fig. 7 A frame with a 640x480 resolution.](image)

VGA Controller, creates the timing information required by the LCD daughter cards or by the on-board VGA DAC.

The VGA Controller generates the timing signals required for the VGA DAC and LCD daughter cards, including horizontal and vertical synchronization signals. The timing information generated by the VGA Controller produces screen resolutions of 800x480 pixels for the VGA DAC and LCD Touchscreen. To generate the timing information correctly, a 50 MHz clock has to be provided to the VGA Controller. This is the default clock in the SOPC Builder software and on the DE boards. In addition, a 25 MHz clock signal must be supplied to the VGA DAC or the LCD daughter cards. For the VGA DAC, this clock can be supplied via the VGA_CLK pin. The External Clocks for DE Board Peripherals core, also provided by the Altera University Program, can generate the required 25 MHz clock; see its documentation for more details.

C. **LCD Design Example:**

Figure 8 shows an example of a typical application that benefits from the improved Cyclone III PLLs. The system drives a LCD display and integrates functions, such as timing controller, video processing, and a memory controller into a single Cyclone III FPGA.

![Fig. 8 LCD Display Design Example](image)
This example takes advantage of several features of the Cyclone III PLL. The first is the ability to drive multiple clocks. This application has multiple clock domains that the FPGA must generate. It must be able to take in the clock frequency from the video source and generate clocks for the memory interface, the internal logic for the FPGA, and any multiple clocks the LCD display may require. Depending on the size and resolution of the LCD, the panel may have multiple clocks that need to be driven. Using the numerous outputs and multiple PLLs will allow you to generate all the clocks needed internally and externally.

The second feature is reconfigurability. In many cases, the clock rates are unknown. The video source may come in different formats with different clock frequencies, or the refresh rate and resolution of the LCD may differ from one panel to another or change on the fly. The wide frequency range of the PLL allows it to accommodate a number of different sources, as well as switch and readjust frequencies on the fly. In other FPGA architectures, the whole FPGA would need to be reconfigured to produce the right clocks, but this reconfiguration takes time and requires special design consideration. The reconfigurability of the Cyclone III PLL insures that this takes place smoothly and with minimum downtime[5].

VI. CONCLUSION

In this paper, we proposed a general methodology that can be exploited for the implementation on a complex System on Programmable Chip and we used a video processing for realization. The results show that the flow provides an effective and low cost approach to the partial dynamic reconfiguration and mixed hardware-software execution problems. Its strength lies on the introduction of the partial dynamic reconfiguration degree of freedom at design time. The proposed flow organizes the input specification into three different components: hardware, reconfigurable hardware and software, managed by proper portion of the methodology.

REFERENCES


[3] “A Simulation Tool For Dynamically Reconfigurable Field Programmable Gate Arrays,” Patrick Lysaght and Jon Stockwood


[10] “Digital Logic and Microprocessor Design With VHDL,” a book by Enoch O. Hwang La Sierra University, Riverside

Cloud Computing and its Security Issue for Building the User Trust

Pawan Kumar
MBU, Solan, Himachal Pradesh
E-mail: pawanjal123@gmail.com

Abstract - Cloud computing is new era computational paradigms and be the innovative business model for organizations that remotely store the information and enjoy the on-demand application services. Cloud computing is a service operation to its customers for data outsourcing and user can be relieved from the burden of local data storage and maintenance. But cloud computing also raises one tremendous issue of “Trust”. So truly implement cloud computing we need to gradually improve it in academic, legal and institutional. The issue of trust is one of the biggest obstacles for the development of cloud computing. The cloud computing customer directly operates the software and OS on the web 2.0 environment. Cloud Computing customer behavior impact and destruction for the software and hardware is much worse than the internet users. Customer trust, principal and technique evaluation are the main issue in future computing.


I. INTRODUCTION

Cloud computing services are springing up like mushrooms after rain, and the entire troupe wants a piece of pie. Cloud computing provides many opportunities for enterprises by offering a wide range of computing services. The current competitive market based environment provides the different service like energy, elasticity, and different choices based services offered by this highly salable technology are too attractive for enterprises and not to be ignore. But these opportunities however, don’t come without challenges. All the benefits of the customers must be secure and based on the trusted environment. As online access to services becomes omnipresent and the cloud access mode gains momentum, authentication is increasingly becoming a main point for security professionals. Cloud computing makes word of service just one click behind their reach of the data like bank accounts, health records, corporate intellectual Property and politically sensitive information. According to the U.S. government expect that the annual growth rate of its spending on cloud computing will be about 40% in the 2010-2015, in 2015 it will reach 7 billion U.S. dollars [1]. The Cloud Computing model has main three types services of namely, Software as a Service (SaaS), Platform as a Service (PaaS) and Cloud Infrastructure as a Service (IaaS).The basic structure of the cloud computing model as shown in Figure 1.1. The cloud service providers(CSP) use the resources provided by resources layer and their technology (such as Virtualization Technology) to integrate the cloud services, and through the information transport layer to provide these services to users[6].

Infrastructure-as-a-service (IaaS): The Cloud service model is based on the virtualization technology. Amazon EC2 is the most IaaS provider where cloud providers deliver computation resources, storage and network as internet-based services.

Platform-as-a-service (PaaS): The cloud computing services can provides the platform to its users where CSP deliver platforms, tools and other business services that enable customers to develop, deploy, and manage their own applications, without installing any of these platforms or support tools on their local be hosted on top of IaaS model or on top of the cloud infrastructures directly. Google Apps and Microsoft Windows Azure are the most known as cloud service providers.

![Figure 1.1 Cloud Service Delivery Model](image-url)
Software-as-a-service (SaaS): The SaaS is cloud providers deliver applications based model that hosted on the network. The cloud infrastructure as internet based service for end users, without requiring installing the applications on the customers’ computers. This model located on the top cloud model of PaaS, IaaS or directly hosted on cloud infrastructure. Salesforce CRM, GoGrid is an example of the provider.

Web 2.0 and Trusted computing: The cloud computing is provides the services through web where customer share their information on the web. But the point of “Trust” is in the mind of user about the web platform. So here user can found the concept of the Trusted computing about there application. Today Web 2.0 was referred as Web as a platform. The web 2.0 aims to make it possible for the people to take responsibility to publish their information on the web, including organizations, businesses and individual users [7].

Security Barriers of Trusted Environment:

In the Cloud Computing Service model no doubt Web2.0 technology is used i.e. for information sharing on the web and i.e. a secure environment for data sharing. But still there is wide variety of security implication for making the trust in the cloud computing of customer. The Cloud service model each layer IaaS, PaaS, SaaS there are some secure requirements that must be full fill for the trust of the cloud users. Some of the implications are the[8]:

1 SaaS security issues: SaaS is that it allows anytime/anywhere access to the software and data. VPns, FTP servers, remote access software, and other cumbersome means of accessing on-premise software are no longer needed.
   a) Information Security
   b) Identity Management
   c) What Polity standards should be taken for the cloud computing.
   d) Data access any where is also a security Issue.
   e) No information about data storage where it is.

2 IaaS security issues: IaaS is popular in the data center where software and servers are purchased as a fully outsourced service and usually billed on usage and how much of the resource is used – compared. Here the issues of security are:
   a) Data leakage protection and usage monitoring [2],
   b) Authentication and authorization
   c) Incident response and forensics capabilities
   d) Infrastructure hardening
   e) End to end encryption

3 PaaS security issues: The PaaS give some control to the people to build applications on top of the platform. In PaaS the security on the application level considered like as host level and network intrusion level and its prevention. The issues of security are:
   a) Less operational control than IaaS
   b) Vendor lock-in
   c) Lack of security tools, reporting from Hackers etc.
   d) Increased privileged user attack likelihood or user dependent.
   e) Cloud provider’s long term viability

Mistrust Reasons on Cloud Computing of Behavior

The above discussion about the cloud service model and their security implications on SaaS, PaaS and IaaS break the trust on the cloud computing of its customer. Cloud computing is basically provides everything as a service just like banks in the govt. sector people deposit their money here and forget about its security for a time and they cannot bother its storage, location. But on the other hand in cloud computing it matters a great cause is that here is our important information, data i.e. is more valuable than money. So we have following reasons of the mistrust[5]:

(1) Unwanted or Malicious software on the cloud outcome decrease trust in user behavior.
(2) Identification and Authentication on the cloud.
(3) Market behavior such as to prove best from others cause is that commercial competitors in the market or against competitors.
(4) Some time user face online configuration errors on the user software applications.
(5) Data Storage and location not known by the cloud user that increases the fear among user about data security.

Customer behavior Evaluation on cloud Computing: In the cloud computing customer can use the services provides by the cloud service provider. So to evaluate the cloud user behavior we must follows the two category in the cloud first is Cloud user(CU) and
other is Cloud service Provider (CSP) so trust belongs to between these two categories [4]. But this is not true concept. For example there is also other resources supplier like electronic resources supplier to the cloud user and CSP. And his behavior influences the customer trust on the cloud. The user behavior can be affected or user also evaluates his services that provides by service providers in the form of different types of applications. So on the basis of the providers there are thee types of providers 1) Internet service provider (ISP) 2) Cloud Service Provider, and 3) Enterprise Service Provider. So all the cloud users use the above service provider services not only the clouds services but local services also. Here customer behaves like evaluator using all these services. Customer can evaluate the cloud service here by the providers user interface processing speed, storage capacity, processing speed and others parameter also.

**Customer Evaluation Principle:** For making the trust on the cloud one of the most important issues for the customer principles over the liability of the Cloud provider. Here pay per use method implemented so basic legal principle is agreement between user and the provider. So this means that the customer and cloud user will take some legal, moral, principal to evaluate the trust behavior of the cloud computing. SLA promised for legal guarantee to provide the uninterrupted services based on the term and conditions. Some Principal are [1]:

1. Evaluating the behavior of the cloud service provider for a average time if there is abnormal activities.
2. Customer trust also depends on the capacity of the provider that fixed number of times of how user access cloud resources smoothly.
4. Punishment of cheating tricks for the cloud provider from the supreme administrator means governance.
5. Exposé of the information when mandated. When exposé is necessary due to a legal or regulatory requirement, then this exposé must be performed.

**Customer Evolution Technique for the building trust in cloud computing**

![Figure 1.2 Customer behavior trust evaluation Techniques of cloud computing](image)

When the new technology came in the market there is only one most important question about that technology that is “TRUST”. So when the customers use that technology for their betterment he must evaluate that. The customer trust evaluation has two types: (a) User behavior based trust (b) Technical or Science based trust. Customer use both of this technology. As figure 1.2 show that cloud computing customer behavior signed by the cloud service providers and users that based on the four layer as mentioned in figure 1.2. User Identity authentication trust refer that when the user authentication may be wrong on the cloud how to re-authenticate the user identification smoothly, for example we are using Amazon EC2 and login with a specific username and password which follows encrypted security but not access the account so how this problem cured on the cloud without loosing out data. Contract behavior trust is refer to whether user behavior comply with the contract, for example, In the use of digital resource cloud, whether the cloud user use resources according to the regulations, whether unwarranted downloading, secretly setting of external proxy server. The cost of digital resource on the cloud another factor of the trust, cloud users are generally paid fixed costs annually for their uses. If the user early quit from system after retrieving information that make him insecure about their account of wrong use. So cloud computing user and the cloud computing service provider both can use the some qualitative and quantitative methods for making the trust of the customers on this new emerging technology but
providing the security is the main work of the Cloud Service Provider.

CONCLUSION

This Paper mainly discusses the cloud computing layers and there security aspects. The evaluation of any technique is importance but the increasing trends of the cloud computing force the customer behavior trust and their evaluation technique for the cloud computing. In this paper I include the barriers of the trust and there analysis, we also discuss the principle that are adopted by the user for evaluating behavior trust, The basic theme of this paper is the what technique opted by the user for there trust.

REFERENCES:


An Efficient Multi-constrained Feasible Path Selection for MANET

S. Baskaran & K. Tamizharasu
Jayam College of Engineering & Technology, Tamilnadu, India
E-mail: jsbaskaran@yahoo.com & Hod_cse@jcut.ac.in

Abstract - Quality-of-service (QoS) routing in an Ad-Hoc network is difficult because the network topology may change constantly and the available state information for routing is inherently imprecise. This paper proposes a method for multi-constrained feasible path selection for MANET. We introduce the composite function which allows OLSR to find the feasible path in case of multiple routes are available to reach the same destination. Combinations of additive and non additive QoS metrics are considered to find the best path. Our simulation results show that the QoS version of the OLSR do improve the Packet Delivery Ratio, Throughput, Normalized control overhead, End-to-End delay and Packet loss but with the increased time complexity.

Keywords: MANET, OLSR (Optimized Link State Routing), Multi-constrained QoS Routing.

I. INTRODUCTION

An Mobile Ad Hoc Network is a set of mobile nodes that are equipped with wireless transmitters and receivers which allow them to communicate each other without the infrastructure. Since there is no infrastructure is used for networking the central administration for the network is not required. Each transmitter has a limited effective range; the nodes within the coverage are able to communicate directly otherwise the nodes are communicated through a multi-hop basis. There are number of applications for this type of network. A group of moving soldiers in a battlefield communicates and coordinates with each other. A group of people with portable computers share their data in a conference room without laying cables between them.

The Optimized Link State Routing Protocol (OLSR) is developed for mobile ad hoc networks which can also be used on other wireless ad-hoc networks. It is a proactive link state protocol, the nodes are maintains the route information in the route table, so routes are available immediately when it is required. The source node will be alerted by the topology whenever the node mobility in the route or changes in the bandwidth then the new route should be identified. Instead of pure flooding to identify the route OLSR uses MPR to reduce the number of the host which broadcasts the information throughout the network. The MPR is a node which is selected such that it covers all nodes that are two hops away. The nodes which are selected as a MPR by some neighbor nodes announce this information periodically in their control messages. In route calculation, the MPRs are used to form the route from a given node to any destination in the network.

A. Neighbour Sensing

Each node periodically broadcast its HELLO messages, containing the information about its neighbors and their link status. Each node must detect the neighbor nodes with which it has a direct and bi-directional link. If a node finds its own address in a HELLO message, it considers the link to the sender node as bi-directional.

B. Multipoint Relay Selection

Every node in the network selects its own set of MPRs. A set of selected one hop neighbour nodes are the MPRs. This covers all the two hop neighbours. MPRs minimize the flooding of the broadcast messages.

Network example for MPR selection

<table>
<thead>
<tr>
<th>Node</th>
<th>1 Hop Neighbours</th>
<th>2 Hop Neighbours</th>
<th>MPR(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A,C,F,G</td>
<td>D,E</td>
<td>C</td>
</tr>
</tbody>
</table>

C. MPR information declaration

A Transmission Control message is sent periodically by each node in the network to declare its MPR selector set. i.e the message contains the list of neighbours who have selected the sender node as multi-point relay. The sequence number associated to this MPR selector is also attached to the list. Each node of the network maintains a topology table, in which it records the information about the topology of the network obtained from the TC messages.

D. Routing Table Calculation

To route the packets in the network each node maintains a routing table in the network. The route entries in the routing table consist of destination address, next-hop address, and estimated distance to destination. The entries are recorded in the table for each destination in the network for which the route is known. The routing table is based on the information contained in the neighbour table and the topology table. Therefore, if any of these tables changed the routing table is recalculated to update the route information.

Quality of Service in MANET is defined as a set of service requirements to be met by the network while transmitting the packets between the source and destination. The measurable specified service attributes are end-to-end delay, bandwidth, packet loss, energy and delay. The QoS metrics can be classified as additive, concave and multiplicative. Bandwidth and delay are additive metrics. Bandwidth and energy are concave metric, while cost, delay, and jitter are additive metrics. Bandwidth and energy are concave in the sense that end-to-end bandwidth and energy are the minimum of all links along the path. The end-to-end delay is an additive constraint because it is the accumulation of all delays of the links along the path. The reliability or availability of a link based on some criteria such as link break probability is a multiplicative metric. Finding the best path subject to two or more additive/concave metrics is a complex problem.

The works done so far in MANET for QoS routing consider either only bandwidth or delay. While other important QoS metrics remained untouched. It is thus obvious that the paths that are discovered by these conventional and limited QoS-aware routing protocols without taking all the important QoS parameters into account are not suitable to transmit large amount of data.

II. RELATED WORKS AND PRELIMINARY

A. Related works


B. Preliminary

A network is modeled by a directed graph $G=(V, E)$ where $V$ is the set of nodes and $E$ is the set of Links. Each link $(i,j) \in E$ is associated with $K$ additive QoS parameters $w_k(i,j), k=1,\ldots,K$, where all the parameters are non-negative. The Given $K$ constraints $c_k, k=1,\ldots, K$ the problem is to find a path $p$ from source node $O$ to a destination node $d$ such that [7]

$$w_k(p) = \sum_{(i,j) \in p} w_k(i,j) \leq c_k \ for \ k = 1,2,\ldots,K$$

III. PROPOSED SYSTEM

A. An Efficient Multi-constrained Feasible Path Selection for MANET

Where the QoS metrics are independent to each other a new composite function $p$ is derived. The composite function $p$ applies the additive and non-additive QoS parameters in Euclidean space with some modifications.

$$f(p) = \frac{1}{\sqrt{K}} \sqrt{\sum_{k} \left( \frac{C_i - w_k}{C_i} \right)^2 + \frac{B_p - B_{min}}{B_p}}$$

Consider $K$ is the total number of additive QoS constraints. $C_i$ is the maximum QoS requirement on path $p$ of the $k^{th}$ additive QoS Constraint. $w_k$ is the actual QoS value of the $k^{th}$ additive constraint along the path $p$, $B_p$ is the available bottleneck bandwidth of path $p$, $B_{min}$ is the minimum bandwidth required for application. Higher the value of the function $f(p)$ implies longer the overall Euclidean distance from constraint points i.e the path is out of constraint values with minimum resources and hence will be the more suitable one. If there are multiple feasible routes for the same destination then the proposed composite function will be selected.

B. Algorithm for MPR Selection

In the given algorithm $G$ is a graph which is illustrated as $G = (V, E)$, $V$ is the set of nodes and $E$ is the set of Links. $N1$ is the first hop neighbour and $N2$ is
An Efficient Multi-constrained Feasible Path Selection for MANET

the second hop neighbour. Node s is the source node which calculates the route and I is an intermediate node. B_i, C_i, and F_i are the bandwidth, non-linear cost and the composite function value of the path from node s to i.

1. Create a empty multipoint relay set, MPR(x) = {} 
2. Select the 1 hop neighbour nodes in N1 as MPR nodes which provide the only path to reach some 2-hop neighbour nodes to the MPR set MPR(x)
3. While the 2 hop nodes in the N2 which are not yet covered by any one node in the MPR(x) set.
   a. Add nodes in N1 to MPR(x) which offer the best feasible 1 hop path in terms of maximum value of p.
   b. Mark the 2 hop neighbour as covered.
4. Repeat the above process until all the 2-hop nodes are reachable via at least one of its MPR Nodes.

C. Algorithm for Path Selection Process

Each node maintains a routing table which allows it to route the packets for other destination in the network.

For, G = (V, E), N1 ⊂ V, N2 ⊂ V

\[ b(p)_{s-1} = \text{bottleneck bandwidth on path } p \]

\[ c(p)_{s-1} = \text{max} \left( \frac{w_k(p)}{c_k} \right) \text{ for } 1 \leq k \leq K \]

\[ f(p)_{s-1} = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} \left( \frac{C_k - w_k}{C_k} \right)^2 + \frac{B_x - B_{\text{min}}}{B_x}\cdot B_x > B_{\text{min}} \]

Step 1: Initially, delete all entries from the routing table
Set B_1 = b(p)_{s-1}, C_1 = c(p)_{s-1} and F_1 = f(p)_{s-1}

Step 2: For all members in N1, If B_i ≥ B_{\text{min}} and C_i ≤ C_1 add i to routing table as the destination node with Bandwidth = B_i, Function value = F_i, hop count = 1.

Step 3: For all members in N2, if there is no entry in the routing table to reach I and B_i ≥ B_{\text{min}} and C_i ≤ C_1 add I to the routing table as the destination node with Bandwidth = B_i, function value = F_i, hop count = 2

Else if (B_i ≥ B_{\text{min}} and F_i > F_{\text{rtable}})
Delete entry to i in routing table
Add i to the routing table as the destination node with bandwidth = B_i, Function value = F_i, hop count = 2

Step 4: For all members in the topology set starting when hop count (h) = 2 and increment it each time by 1
If there is no entry in the routing table to reach I and B_i ≥ B_{\text{min}} and C_i ≤ C_1
Add i to the routing table as the destination node with bandwidth = B_i, Function value = F_i, hop count = h + 1
Else if (B_i ≥ B_{\text{min}} and F_i > F_{\text{rtable}})
Delete entry to i in routing table
Add i to routing table as the destination node with Bandwidth = B_i, Function value = F_i, hop count = h + 1

D. Simulation Model and Parameters

For simulating the original OLSR protocol and the modified OLSR protocol we have used the OLSR protocol implementation which runs in version 2.29 of the Network Simulator NS2[9]

<table>
<thead>
<tr>
<th>TABLE I. SIMULATION PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
</tr>
<tr>
<td>Software</td>
</tr>
<tr>
<td>MAC Protocol</td>
</tr>
<tr>
<td>Node’s Transmission range</td>
</tr>
<tr>
<td>Topology area</td>
</tr>
<tr>
<td>Number of nodes</td>
</tr>
<tr>
<td>Mobility Model</td>
</tr>
<tr>
<td>Packet Size</td>
</tr>
<tr>
<td>Traffic type</td>
</tr>
<tr>
<td>Link bandwidth</td>
</tr>
<tr>
<td>Routing protocol</td>
</tr>
<tr>
<td>Hello and TC message interval</td>
</tr>
<tr>
<td>Simulation time</td>
</tr>
<tr>
<td>Node mobility speed</td>
</tr>
</tbody>
</table>

E. Performance Evaluation Metrics

The performance evaluation aims to assess the improvement achieved in Modified OLSR to find the best path. The evaluation is done using the various metrics like

Average End-to-End delay: The average time required to send the packets to source to destination. The delay parameter will be calculated to improve the selection of best path by the proposed routing algorithm. The average delay is calculated by subtracting “time at which first packet was transmitted by source” from “time at which first data packet arrived to destination”. This includes all possible delays caused by buffering...
during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, propagation and transfer times. This metric is crucial in understanding the delay introduced by path discovery.

**Packet Delivery Ratio:** This will be calculated based on number of packets received by the destination divided by number of packets sent by source node. This performance metric will give us an idea of how well the protocol is performing in terms of packet delivery at different speeds using different traffic models.

**Normalized Control Overhead:** Ratio of the total number of control overheads to the total number of packets received

Figure 2: Packet Delivery Ratio (PDR)

Here, we analyze the average number of routing packets required to deliver a single data packet. This metric gives an idea about the extra bandwidth utilized by overhead to deliver data traffic.

**Throughput:** The amount of data received by the destination during the simulation period. It can be calculated of delivered data packets divided by the total duration of simulation time. We analyze the throughput of the protocol in terms of number of messages delivered per one second.

**Packet Loss:** More packet losses may occur in MANET than wired network. The dynamic nature of MANET topology and unpredictable movement of the mobile node imposes a great challenge in ensuring the error free transmission. Packet losses caused by transmission failures, link failures, and network congestion coexist. In MANET, Links fail frequently leading to packet loss or delay in transmission.

**CONCLUSION**

In this work we proposed a method to find the multi-constrained feasible path in MANET utilizing the maximum resources available with the nodes. The extended OLSR is made to find the feasible path, because the original OLSR does not support QoS. In case of multiple paths are available to the destination, our composite function will judge the more suitable path. According to the simulation results, our proposed algorithm proves the improvement over the original OLSR protocol with additional time complexity.
REFERENCES


Classification and Prediction in Customer Relationship Management Using Back Propagation

Leela Rani KommaReddy & G Loshma
Sri Vasavi Engineering College, Pedatadepalli, Tadepalligudem—534101, India
Email: leela.rani2@gmail.com & loshma@gmail.com

Abstract - Customer Relationship Management provides a customer classification and prediction which is used for the optimization of business process. The classification and prediction which is used for the optimization of business process. This classification and prediction in CRM will help the company to study, analyze and forecast customers pattern of consumption, business transaction and purchasing CRM has become major activity in the enterprise based business organization using the CRM. CRM is an important activity in the enterprise business organization like banking industry, insurance industry, retail industry and manufacture industry. In the system we are using data mining techniques to implement customer classification in CRM as we need to analyze mass volume of data we are implementing an efficient and effective Neural Network based technique. Based on the existing system like Naïve Bayesian System, our proposed system implements Back propagation Neural Network techniques which would generate accurate results with less time complexity.

Keywords— Classification, Prediction, Data Mining, Naïve Bayesian Algorithm, Back propagation, Neural networks.

I. INTRODUCTION

The core part of CRM activities is to understand customer requirements and retain profitable customers. To reach it in a highly competitive market, satisfying customer’s needs is the key to business success [1]. Unprecedented growth of competition has raised the importance of retaining current customers. Retaining existing customers is much less expensive and difficult than recruiting new customers in a mature market. So customer retention is a significant stage in Customer Relation Management, which is also the most important growth point of profit [3]. Factors that influence customer satisfaction degree are concerned by all enterprise managers [9]. Marketing literature states that it is more costly to engage a new customer than to retain an existing loyal customer. Churn prediction models are developed by academics and practitioners to effectively manage and control customer churn in order to retain existing customers [4]. So, Customer satisfaction is most important.

Data mining (DM) methodology has a tremendous contribution for researchers to extract the hidden knowledge and information which have been inherited in the data used by researchers [7]. Data mining has a tremendous contribution to the extraction of knowledge and information which have been hidden in a large volume of data [8]. The concept of customer satisfaction and loyalty (CS&L) has attracted much attention in recent years. A key motivation for the fast growing emphasis on CS&L can be attributed to the fact that higher customer satisfaction and loyalty can lead to stronger competitive position resulting in larger market share and profitability [6][7].

However, it is a difficult and a complex task to identify the customer’s needs such as colors and design of the products. The objective of this paper is to design and implement the expert system in order to assess customer satisfaction and reveal appropriate strategies to improve it. As the customer satisfaction on colors and design can have a complex hidden pattern and, therefore, the approach of the paper should have an ability to perform pattern recognition, classification and forecast which make the artificial neural networks an appropriate technique to be applied in the expert system[9]. The conceptual work of the paper is illustrated in Figure 1, in which the assumption of the customer requirements and expert system are based upon the statement that “in general, the same customer group will like the same colors”. A vast variety of colors mixing in different products that makes it a difficult and complicated task to identify the customer’s needs. The contribution of this paper is in designing the system that is the combination of the expert system and the ANN. The customers can interact with the interface of the expert system to ask and get the advices from the system. Correlation Coefficient can be found. According to that, we can identify the customer’s behavior [8][9].

International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati
II. LITERATURE REVIEW

In the process of economic development, business management concept evolved from product-oriented idea to market-oriented idea, and then to customer-oriented idea. Whether enterprises can obtain, maintain and develop their own clients or not has become the most critical factor because customers are important strategic resources. Customer Relationship Management (CRM) is based on the respect to customers and it can help enterprises understand the entire life cycle of the customers and enable enterprises to provide the customers with more personalized and more efficient services, and then it can enhance customer satisfaction and loyalty, and improve the competitiveness of enterprises ultimately. Customer segmentation, as the core of CRM, is to classify the customers according to the customer’s attributes, behavior, needs, preferences, value and other factors in a clear business strategy and specific market, and it can provide appropriate products, services and marketing models to the customers.

A. Data Mining and CRM

The first and simplest analytical step in data mining is to describe the data. For example, you can summarize data’s statistical attributes (such as means and standard deviations), visually review data using charts and graphs and look at the distribution of field values in your data. But data description alone cannot provide an action plan. You must build a predictive model based on patterns determined from known results and then test that model on results outside the original sample[1][2]. A good model should never be confused with reality (you know a road map isn’t a perfect representation of the actual road), but it can be a useful guide to understanding your business. Data mining can be used for both classification and regression problems. In classification problems you’re predicting what category something falls into – for example, whether or not a person is a good credit risk or which of several offers someone is most likely to accept. In regression problems, you’re predicting a number, such as the probability that a person will respond to an offer. In CRM, data mining is frequently used to assign a score to a particular customer or prospect indicating the likelihood that the individual behaves the way you want [1][2][3]. For example, a score could measure the propensity to respond to a particular offer or to switch to a competitor’s product. It is also frequently used to identify a set of characteristics (called a profile) that segments customers into groups with similar behaviors, such as buying a particular product. A special type of classification can recommend items based on similar interests held by groups of customers. This is sometimes called collaborative filtering.

B. Expert System

The expert system’s role is in the preparation to capture the knowledge of the experts and the data from the customer’s requirements. The system has the capability to compile the collected data and form the appropriate rules for choosing fragrance notes for the products. In order to identify the hidden pattern of the customer’s needs, the artificial neural networks technique has been applied to classify the fragrance notes based upon a list of selected information [3]. The expert system’s role is in the preparation to capture the data from the customer’s requirements and predict appropriate perfume. For this end, factors of perfume customers’ decision were recognized using Fuzzy Delphi method and a back propagation neural network classification model was developed and trained with 2303 data of customers [4]. The proposed business intelligent system for demand forecasting proves to give more accurate prediction for future demands compared to the existing models and practices in spare parts inventory management. This helps inventory managers to better manage their supply chain performance by reducing reaching days and service level simultaneously. Reaching day as a measure of inventory level is generally reduced successfully by the retailers at the cost of service level in most of the places [5].

C. Bayesian Classification

Bayesian classification is based on Bayes theorem.
Studies comparing classification algorithms have found that simple Bayesian classifier known as the naive Bayesian classifier to be comparable in performance with decision and neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. The naïve Bayesian classifier works as follows, as in [7]:

1. Each data sample is represented by an n-dimensional feature vector \( X = (x_1, x_2, ..., x_n) \), depicting \( n \) measurements made on the sample from \( n \) attributes, respectively, \( A_1, A_2, ..., A_n \).

2. Suppose that there are \( m \) classes, \( C_1, C_2, ..., C_m \). Given an unknown data sample, \( X \) (having no class label), the classifier will predict that \( X \) belongs to the class having the highest 3 posterior probability, conditioned on \( X \). That is, the naïve Bayesian classifier assigns an unknown sample \( X \) to the class \( C_i \) if and only if \( P(C_i | X) > P(C_j | X) \) for \( 1 \leq j \leq m, j \neq i \). Thus we maximize \( P(C_i | X) \). The class \( C_i \) for which \( P(C_i | X) \) is maximized is called the maximum posterior hypothesis. By Bayes theorem, \( P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)} \) for \( 1 \leq i \leq m \). Otherwise, we maximize \( P(X | C_i)P(C_i) \).

3. As \( P(X) \) is constant for all classes, only \( P(X | C_i)P(C_i) \) need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, \( P(C_1) = P(C_2) = ... = P(C_m) \) and we would therefore maximize \( P(X | C_i)P(C_i) \).

4. Given data sets with many attributes, it would be extremely computationally expensive to compute \( P(X | C_i)P(C_i) \). In order to reduce computation in evaluating \( P(X | C_i) \), the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the sample, that is, there are no dependence relationships among the attributes. This can be explained using an Example as follows.

Suppose commercial banks hope to increase the number of valuable customer information in huge amounts of data accumulated by commercial banks, which is used to identify customers and provide decision support. We wish to predict the class label of an unknown sample using naive Bayesian classification, given the training data as Table 1. The data samples are described by the attributes: sex, age, student and income. The class label attribute, creditcard_proposing has two distinct values (namely, \{yes, no\}).

### III. SYSTEM ARCHITECTURE

If data warehouse, OLAP and other data analytic tools are directly applied to enterprise's data base, for example sale data base, marketing data base and customer service data base, it will be not only waste time and substance but also not ideality, therefore, it is wise that the data of business data base is lead to data warehouse after was cleaned, extracted, transformed, loaded, the data of data warehouse will offer the best data resource for the application of data mining. The data analysis will become great efficiency based on the data warehouse. The architecture of CRM based on data warehouse is shown in Figure 3.1.

**IV. PROPOSED WORK (BACK PROPOGATION BASED CLASSIFICATION)**

#### A. Classification

Classification is a data mining (machine learning) technique used to predict group membership for data instances. For example, you may wish to use classification to predict whether the weather on a particular day will be “sunny”, “rainy” or “cloudy”. Popular classification techniques include decision trees and neural networks. It can be formally defined as given a collection of records (training set)

- Each record contains a set of attributes, one of the attributes is the class.

Find a model for class attribute as a function of the values of other attributes.

Goal: previously unseen records should be assigned a class as accurately as possible.

- A test set is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

The following diagram illustrates the classification task. Figure 4.1
Classification Process.

B. Back Propagation

Multilayer perception I & BP (Back-propagation) model Standard multilayer perception (MLP) architecture consists more than 2 layers; A MLP can have any number of layers, units per layer, network inputs, and network outputs such as fig 4.2 models. This network has 3 Layers; first layer is called input layer and last layer is called output layer; in between first and last layers which are called hidden layers. Finally, 4 this network has three network inputs, one network output and hidden layer network.

Fig4.2 Standard Multi layer perception architecture

This model is the most popular in the supervised learning architecture because of the weight error correct rules. It is considered a generalization of the delta rule for nonlinear activation functions and multilayer networks. In a back-propagation neural network, the learning algorithm has two phases. First, a training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output pattern is generated by the output layer. If this pattern is different from the desired output, an error is calculated and then propagated backward through the network from the output layer to the input layer. The weights are modified as the error is propagated. According to the Richard P. Lippmann, he represents step of the back-propagation training algorithm and explanation. The back-propagation training algorithm is an iterative gradient designed to minimize the mean square error between the actual output of multi-layer feed forward perception and the desired output. It requires continuous differentiable nonlinearity. The following assumes a sigmoid logistic nonlinearity.

Step1: Initialize weights and offsets
Set all weights and node offsets to small random values.

Step2: Present input and desired outputs Present a continuous valued input vector X0, X1,...,XN1 and specify the desired output d0,d1,...,dM-1. If the net is used as a classifier them all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The input could be new on each trial or samples from a training set could be presented cyclically until stabilize.

Step 3: Calculate Actual Output Use the sigmoid nonlinearity from above and formulas as in fig 3 to calculate output y0,y1,...,yM-1.

Step 4: Adapt weights Use a recursive algorithm starting at the output nodes and working back to the first hidden
layer. Adjust weights by \( W_{ij}(t+1) = W_{ij}(t) + n \delta_j x_i \) - (1)

In this equation \( W_{ij}(t) \) is the weight from hidden node \( i \) or from an input to node \( j \) at time \( t \), \( \delta_j \) is either the output of node \( i \) or is an input, \( \eta \) is a gain term, and \( \delta_j \) is an error term for node \( j \), if node \( j \) is an output node, then

\[
\delta_j = y_j(1-y_j)(d_j-y_j) \quad -(2)
\]

where \( d_j \) is the desired output of node \( j \) and \( y_j \) is the actual output.

Step 5: Repeat by going to step 2

V. IMPLEMENTATION DETAILS

In this section, we analyze the performance of our new approach for classifying the customer credit card data using back propagation approach and predicting the who likely customers to our new business are. This we compare with Bayesian classification approach. The Algorithms were implemented in DOT NET Framework using C# language. Forms framework is used for designing GUI. We have placed transactional data records in data sets. The SQL Server 2000 data base is used for managing the performance results.

VI. EXPERIMENTAL RESULTS

In order to evaluate the performance of our proposed algorithm, we have conducted experiments on a PC (CPU: Intel(R) Core2 Duo, 3.16GHz) with 4GByte of main memory running Windows XP. The following shows the results of Back propagation Classification Algorithm for generating customer relationship prediction.

Fig 6.1 Customer Profile Updation 5

Fig 6.2 Preparing training Data

Fig 6.3 Naïve Bayesian Classification of Data

Fig 6.4 Back Propagation Classification of data.
VII. CONCLUSION AND FUTURE WORK

In the dynamic business environment, information systems need to evolve to adapt to new requirements, which may be driven by CRM. The paper has presented a framework of an evolving information system based on knowledge from data mining, and has discussed the framework by focusing on knowledge of classification. Data mining provides the technology to analyze mass volume of data and detect hidden patterns in data to convert raw data into valuable information. This paper mainly focused on the research of the customer classification and prediction in Customer Relation Management concerned with data mining based on Back propagation technique.

REFERENCES

Self Configurable Re-Link Establishment Using Continuous Neighbour Discovery In Asynchronous Wireless Sensor Networks

M. Sreedevi¹, Nadilla Khadar Valli², R. Seshadri³

¹ C.S.E, MITS, Madanapalle, A.P, ² MITS, Madanapalle, A.P
³ Computer Center, S.V.U, Tirupati, A.P

Abstract - In many sensor networks the nodes are static and node connectivity is matter to changes because of disturbance in wireless communication, transmission power varies, or loss of synchronization between neighbouring nodes. Hence, even after a sensor is aware of its immediate neighbours, it must continuously maintain its view, a process we call continuous neighbour discovery. In this work we distinguish between neighbour discovery during sensor network initialization and as well as in it’s network life time. We focus on the latter and view it as a joint task of all the nodes in every connected segment. Each sensor employs a simple protocol in a coordinate effort to reduce power consumption without increasing the delay required to detect hidden sensors in the network.

Keywords: Neighbour discovery mechanisms, network life time, delay, static nodes, sensor activity.

I. INTRODUCTION

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. In the sensor network model considered in this paper, the nodes are placed randomly over the area of interest for providing efficient communication each sensor node need to detect their immediate neighbours. However, for sensor networks with low and irregular traffic, a special neighbour discovery scheme should be used. This paper mainly analyzes the existing neighbour discovery scheme and resources such as energy, memory, computational speed and communications bandwidth of wsn.

In most sensor networks the nodes are static. Nevertheless, node connectivity is subject to changes because of disruptions in wireless communication, transmission power changes, or loss of synchronization between neighbouring nodes

When we compare with traditional wireless communication networks like MANET’s and cellular networks the WSN’s are having unique characteristics such as high energy computation, huge number of node deployment and some of the storage capability [1], communication i.e. continuous neighbour discovery. Which discussed so many number of new challenging in the communication, development & deployment of the WSNs? In order to solve the various design and application issues a huge amount of research is already carried out. Different continuous neighbour discovery schemes were proposed for efficient communication of a network and routing the fact data to the base station [9],[10][11][12][13][14].

In sensor networks Communication is totally depending upon the application. Hence, even after a sensor is aware of its immediate neighbours, it must continuously maintain its view; the sensors must continuously look for new neighbours in order to accommodate the following situations:

- Loss of local synchronization due to accumulated clock drifts;
- Disruption of wireless connectivity between adjacent nodes by a temporary event, such as a passing car or animal, a dust storm, rain, or fog; when these events are over, the hidden nodes must be rediscovered
- The on going addition of new nodes, in some network, to compensate for nodes that have ceased...
to function because their energy has been exhausted.

- To increase in the transmission power of some nodes, in response to certain events, such as detection of emergency situations.

For these reasons, detecting new links and nodes in sensor networks must be considered as an ongoing process. When the sensor performs continuous neighbour discovery, it is already aware of most of its immediate neighbours and can therefore perform it together with these neighbours in order to consume less energy. In contrast, initial neighbour discovery must be executed by each sensor separately.

In Fig. 1 shows a typical neighbour discovery protocol. In this protocol, a node becomes active according to its duty cycle. Let this duty cycle be $\alpha$ in Init state $\beta$ in normal state. We want to have $\beta < \alpha$ .When a node becomes active; it transmits periodical HELLO messages and listens for similar message from possible neighbours. A node that receives a HELLO message immediately responds and the two nodes can invoke another procedure to finalize the setup of their joint wireless link.

Fig. 1. Transmission of HELLO messages in Init and Normal states

To summarize, in the Init state, a node has no information about its surroundings and therefore must remain active for a relatively long time in order to detect new neighbours. In contrast, in the Normal state the node must use a more efficient scheme. Such a scheme is the subject of our study. Fig. 2 summarizes this idea. When node is in the Init state, it performs initial neighbour discovery. After a certain time period, during which the node is expected, with high probability, to find most of its neighbours, the node moves to the Normal state, where continuous neighbour discovery is performed. A node in the Init state is also referred to in this paper as a hidden node, and a node in the normal state is referred to as a segment node.

The main idea behind the continuous neighbour discovery scheme we propose is that the task of finding a new node $u$ is divided among all the nodes that can help $v$ to detect $u$. These nodes are characterized as follows:

- They are also neighbors of $u$.
- They belong to a connected segment of nodes that have already detected each other.
- Node $v$ also belongs to this segment.

Let $\text{degs}(u)$ be the number of these nodes. This variable indicates the in-segment degree of a hidden neighbour. In order to take advantage of the proposed discovery scheme node $v$ must estimate the value of $\text{degs}(u)$.

Related work:-

In a Wi-Fi network operating in centralized mode, a special node, called an access point, coordinates access to the shared medium. Messages are transmitted only to or from the access point. Therefore, neighbour discovery is the process of having a new node detected by the base station. Since energy consumption is not a concern for the base station, discovering new nodes is rather easy. The base station periodically broadcasts a special HELLO message. A regular node that hears this message can initiate a registration process. The regular node can switch frequencies/channels in order to handle the best HELLO message for its needs. Which message is the best might depend on the identity of the broadcasting base station, on security considerations, or on PHY layer quality (signal-to-noise ratio). Problems related to possible collisions of registration messages in such a network are addressed in [4]. Other works try to minimize neighbour discovery time by optimizing the broadcast rate of the HELLO messages [1], [5], [6], [7], [8]. The main differences between neighbor discovery in
WiFi and in mesh sensor networks are that neighbor discovery in the former are performed only by the central node, for which energy consumption is not a concern. In addition, the hidden nodes are assumed to be able to hear the HELLO messages broadcast by the central node. In contrast, neighbor discovery in sensor networks is performed by every node, and hidden nodes cannot hear the HELLO messages when they sleep. In mobile ad-hoc networks (MANETs), nodes usually do not switch to a special sleep state. Therefore, two neighboring nodes can send messages to each other whenever their physical distance allows communication. AODV [9] is a typical routing protocol for MANETs. In AODV, when a node wishes to send a message to another node, it broadcasts a special RREQ (route request) message. This message is then broadcast by every node that hears it for the first time.

The same message is used for connectivity management, as part of an established route maintenance procedure, aside from which there is no special neighbor discovery protocol. Minimizing energy consumption is an important target design in Bluetooth [10]. As in Wi-Fi, the process of neighbor discovery in Bluetooth is also asymmetric. A node that wants to be discovered switches to an inquiry scan mode, whereas a node that wants to discover its neighbors enters the inquiry mode. In the inquiry scan mode, the node listens for a certain period on each of the 32 frequencies dedicated to neighbor discovery, while the discovering node passes through these frequencies one by one and broadcasts HELLO in each of them. This process is considered to be energy consuming and slow. A symmetric neighbor discovery scheme for Bluetooth is proposed in [11]. The idea is to allow each node to switch between the inquiry scan mode and the inquiry mode.

II. PROBLEM DEFINITION:-

We assume that all nodes have the same transmission range, which means that connectivity is always bidirectional. During some parts of our analysis, we also assume that the network is a unit disk graph; namely, any pair of nodes that are within transmission range are neighboring nodes. Two nodes are said to be directly connected if they have discovered each other and are aware of each other's wake-up times. Two nodes are said to be connected if there is a path of directly connected nodes between them. A set of connected nodes is referred to as a segment. Consider a pair of neighboring nodes that belong to the same segment but are not aware that they have Direct wireless connectivity.

Fig 3. Segments with hidden nodes and links

In fig 3.(a) the nodes and c can learn about their hidden wireless link using the following simple scheme, which uses two message types:

(a) SYNC messages for synchronization between all segment nodes, transmitted over known wireless links;
(b) HELLO messages for detecting new neighbours.

Scheme 1 (detecting all hidden links inside a segment):

This scheme is invoked when a new node is discovered by one of the segment nodes. The discovering node issues a special SYNC message to all segment members, asking them to wake up and periodically broadcast a bunch of HELLO messages. ThisSYNC message is distributed over the already known wireless links of the segment. Thus, it is guaranteed to be received by every segment node. By having all the nodes wake up almost at the same time, for a short period, we can ensure that every wireless link between the segment's members will be detected.

Scheme 2 (detecting a hidden link outside a segment):

A random wake-up approach is used to minimize the possibility of repeating collisions between the HELLO messages of nodes in the same segment. Theoretically, another scheme may be used, where segment nodes coordinate their wake-up periods to prevent collisions and speed up the discovery of hidden nodes. However, finding an efficient time division is equivalent to the well-known node colouring problem, which is Node u wakes up randomly, every T(u) seconds on the average, for a fixed period of time H. During this time it broadcasts several HELLO messages, and listens for possible HELLO messages sent by new neighbours. The value of T(u) is as follows:

• T(u) = TI, if node u is in the Init state
• T(u) = TN(u), if node u is in the Normal state
Proposed method:-

As already explained, we consider the discovery of hidden neighbours as a joint task to be performed by all segment nodes. To determine the discovery load to be imposed on every segment node, namely, how often such a node should become active and send HELLO messages, we need to estimate the number of in-segment neighbours of every hidden node \( u \), denoted by \( \text{degS}(u) \).

In this section we present methods that can be used by node \( v \) in the Normal (continuous neighbour discovery) state to estimate this value. Node \( u \) is assumed to not yet be connected to the segment and it is in the Init (initial neighbour discovery) state. Three methods are presented: 1) Node \( v \) measures the average in-segment degree of the segment's nodes, and uses this number as an estimate of the in-segment degree of \( u \). The average in-segment degree of the segment's nodes can be calculated by the segment leader. To this end, it gets from every node in the segment a message indicating the in-segment degree of the sending node, which is known due to Scheme 1. We assume that the segment size is big enough for the received value to be considered equal to the expected number of neighbours of every node. 2) Node \( v \) discovers, using Scheme 1, the number of its in-segment neighbours, \( \text{degS}(v) \), and views this number as an estimate of \( \text{degS}(u) \). This approach is expected to yield better results than the previous one when the degrees of neighbouring nodes are strongly correlated. 3) Node \( v \) uses the average in-segment degree of its segment's nodes and its own in-segment degree \( \text{degS}(v) \) to estimate the number of node \( u \)'s neighbours. This approach is expected to yield the best results if the correlation between the in-segment degrees of neighbouring nodes is known. An interesting special case is when the in-segment nodes are uniformly distributed.

REFERENCES


Neighbourhood Load Routing and Multi-channels in Wireless Mesh Networks

Nirmala Y.Barker#1 & Poornima. B#2

#Department of CSE, BIET, Davangere, Karnataka
E-mail: nimms.yb@gmail.com, poornima_teju@rediffmail.com:

Abstract - As an emerging technology, wireless mesh networks are making significant progress in the area of wireless networks in recent years. Routing in Wireless Mesh Network (WMN) is challenging because of the unpredictable variations of the wireless environment. Traditional mechanisms have been proved that the routing performance would get deteriorated and ideal metrics must be explored. Most wireless routing protocols that are currently available are designed to use a single channel. The available network capacity can be increased by using multiple channels, but this requires the development of new protocols specifically designed for multi-channel operation. In this paper, we propose Neighbourhood load routing metric in single channel mesh networks and also present the technique to utilize multiple channels and multiple interfaces between routers for communication. The traditional routing metrics Hop Count and Weighted Cumulative Expected Transmission Time (WCETT) are used in routing. We compare performance of AODV-HOP, WCETT and NLR routing metrics in single channel and multi-channel environment by considering throughput and end to end delay performance metrics. Our results show that NLR performs better in single channel environment.

Keywords — Routing Metric, WMN, AODV, WCETT, ETX, ETT, WCETT, NLR.

I. INTRODUCTION

Wireless Mesh Network (WMN) [1] is a promising technology to offer a list of benefits in constructing next generation networks in a sizable geographic area. It brings a lot of benefits to build wireless metropolitan networks as the most outstanding characteristic of WMNs is low cost. It is because instead of using optical fibre cable, wireless radios are applied in WMNs which have been already deployed to build wireless broadband network in some newly developing areas worldwide and isolated islands. A WMN combines the characteristics of both fixed network and MANET. The communication inside a WMN is similar to MANET, client nodes are self-configured and self-organized where the routes are selected by using certain routing algorithm and each client node has to relay other's packets. For accessing backbone internet, the packets are forwarded through internet gateway to the fixed network by fixed cable links.

WMNs can be divided into three main types: Infrastructure, Client, and Hybrid. In an Infrastructure WMN, Mesh Clients gain access to each other or to the backhaul network through Mesh Routers and are not actively involved in routing and forwarding of packets. Thus, all Mesh Clients gain access to Mesh Routers via a single wireless hop. In Client WMNs, Mesh Clients communicate with each other directly, without involving any Mesh Routers. A Client WMN is essentially a pure multi-hop mobile ad-hoc wireless network. A Hybrid WMN combines the connectivity pattern of both the Infrastructure and Client WMNs as shown in Figure 1. In these networks, both Mesh Clients and Mesh Routers are actively involved in routing and forwarding of packets and Mesh Clients can access the wireless backhaul network via multiple client hops. The routing capabilities of clients provide improved connectivity and coverage inside the WMN. The hybrid architecture provides full advantage of the WMN.

A routing metric is implemented in the routing protocol to judge the superiority of a route over other alternate ones. The routing metrics cover a set of routing constraints such as bandwidth, network delay, path length, load balancing, reliability, and communication cost and so forth. In addition, the improvement of one aspect normally results in all other aspects. As mentioned in [1], new routing metrics are required to examine and improve the performance of WMNs in dealing with more constraints. Hence, the design of routing metrics is very important to improve the overall performance of WMNs and MANETs.

Figure 1: The network structure of a typical WMN
For meeting the ever-increasing throughput demands of applications, it is necessary to utilize the entire available spectrum. Multiple channels have been utilized in infrastructure-based networks by assigning different channels to adjacent access points, thereby minimizing interference between access points. However, typical multi-hop wireless network configurations have used a single channel to ensure all nodes in the network are connected. For the full utilization of available channels, it is desirable to have different nodes communicating (in parallel) on different channels. However, when using multiple channels, two adjacent nodes can communicate with each other only if they have at least one interface on a common channel. Therefore it may be necessary to periodically switch interfaces from one channel to another to enable different nodes to communicate with each other. Furthermore, interface switching may have to be carefully coordinated to allow any adjacent pair of nodes to communicate with each other.

Load balancing of WMN becomes a hot topic as it provides better QoS provision in offering high throughput, high packet delivery ratio, low delay, and low jitter. Therefore authors are proposing multichannel assignment for WMN and usage of WCETT metric in routing.

The remainder of this paper is organised as follows. Section II examines routing metrics and usage of multiple channel assignment and examines performance of HOP-COUNT and WCETT routing metrics in multi-channel environment. Section IV presents the simulation results and performance evaluation. Section V concludes our work.

II. RELATED WORK

In this section, we describe series of existing routing metrics, and then show how they work, focusing on their abilities to satisfy the requirements of WMN.

HOP COUNT: It is widely used in existing protocols such as AODV [13], DSR [14], and DSDV [15]. It helps a routing protocol to avoid long transmission paths in finding the routing path with the minimum distance, i.e. hop number. Other issues such as interference, transmission rate, and packet loss ratio are not considered in this routing metric. Therefore, HOP COUNT may result in poor performance in some network environments.

LOAD COUNT [9] [10]: It is a load balancing metric for wireless networks

\[
\text{Load}_i = \sum_{i=1}^{n} \text{Load}_i
\]

where \( \text{Load}_i \) is the traffic load on a node \( i \) which is normally captured using IFQ length. The IFQ (Network Interface Queue) is a drop-tail buffer at the MAC layer of 802.11 radios, which contains outbound frames to be transmitted by the physical layer where the size of IFQ is calculated as the number of remaining packets in the buffer.

Expected Transmission Count (ETX): It is a metric to estimate the expected number of MAC layer transmissions for the wireless links and measure the packet loss rate which is proposed by De Couto et al. [11] [12]. A node sends out probe packets to all its neighbour nodes every second. When a neighbour node receives probes, it increments the amount of received packets and calculates the loss rate of packet every 10 seconds. The weight of a route is the sum of the ETX of all links along the path. The possibility of successful packet transmission from source \( a \) to destination \( b \) in a wireless link is: \( p = (1 - pf) \times (1 - pr) \) Then ETX can be achieve as

\[
\text{ETX} = \sum_{k=1}^{∞} kp^k 1 - p^{k-1}
\]

Where \( pf \) is the probability of successful forwarded packets and \( pr \) denotes the probability of successful received packets. The advantages of \( ETX \) are the reduced probing overhead and non self-interference as the delay is not measured. However, \( ETX \) cannot measure the cause of data size in the delivery ratio and it does not consider the transmission rate. Furthermore, unicast probing of \( ETX \) is not accurate as differences between broadcast and unicast.

ETT (Expected Transmission Time) [18]: Though the expected transmission count matters, the time taken for transmission affects throughput, so ETT was developed and selected as link metric. ETT captures the data rate at which packets are transmitted, means the time required for successful sending of one data packet. In other words, ETT, as metric that improved ETX method, is a method of calculating transmission expected value in MAC layer by reflecting link bandwidth and packet size

\[
\text{ETT} = \text{ETX} \times \text{S} \quad (2)
\]

\( S \) stands for data packet size, and \( R \) the data transmission rate of that link. ETT produces accurate wireless link quality by reflecting transmission rate, but cannot reflect other important problems like protocol overhead and inter-flow interference.

Weighted Cumulative ETT (WCETT): It is also proposed by Draves et al [16] and it considers the multi-radio nature of the WMNs in two components: the total transmission time along all hops in the WMN and the channel diversity in the path. The WCETT of a path \( p \) is
Neighbourhood Load Routing and Multi-channels in Wireless Mesh Networks

WCETT (r) = (1 − p)ETT_i + \frac{P \max_{i\leq j} X_j}{\sum_{i\leq j} k}

where \( p \) is a parameter, \( 0 \leq p \leq 1 \). And path \( r \) uses \( X_j \) number of times of channel \( j \). Therefore, \( \frac{P \max_{i\leq j} X_j}{\sum_{i\leq j} k} \) denotes the maximum number of times that the same channel \( j \) is used along a path. Although it captures the intra-flow interference of a path with measuring the channel assignment time, it does not consider the inter-flow interference. Thus, traffic flows may be routed to the dense area by WCETT. One more important problem of the WCETT is that it is not isotonic which generates a forwarding loop while chosen a path.

III. MULTIPLE CHANNELS AND MULTIPLE INTERFACES

In this section we study the use of multi interface and multi channel in wireless networks.

Many researchers have proposed the use of multiple channels and multiple interfaces for ad-hoc wireless networks [7]. However, we have studied P. Kyasanur’s and N. H. Vaidya’s interface assignment scheme in [2] for our implementation, as this approach is more flexible and versatile among others. Below we present a brief comparison between various related work and [2].

For instance, A. Raniwala, K. Gopalan, and T. Chiueh propose the centralized channel assignment and routing solution in [3]. Unlike this solution that is designed for use in static networks where traffic is directed towards specific gateway nodes, the approach in [2] allows any node to communicate with any other. Additionally, the approach in [3] expects stationary nodes and traffic load on every link, since this information is used to assign interfaces and compute routes. Oppositely, the solution in [2] does not need this information and thus is more suitable for ad-hoc networks involving mobile nodes.

Next, S. Wu, C. Lin, Y. Tseng, and J. Sheu propose a MAC layer solution in [4] that requires two interfaces: one for control, therefore assigned to a common channel; the other for data exchange, thus is switched among the remaining channels. Also, R. Maheshwari, H. Gupta, and S. R. Das propose new MAC protocols for multi-channel operation in wireless ad-hoc and mesh networks in [5]. The aforementioned two proposals all require changes to the existing IEEE 802.11 standard. On the contrary, the approach in [2] can be implemented using existing IEEE 802.11 wireless network interface cards.

Finally, paper [6] proposes a new routing metric called Weighted Cumulative Expected Transmission Time (WCETT), for multi-channel ad-hoc networks. WCETT ensures channel diverse routes are selected by assuming the number of interfaces per node is equal to the number of channels used by the network. In contrast, the interface switching technique proposed in [2] can allow the number of interfaces to be smaller than the number of available channels, while still manages to utilize all the channels.

IV. PROPOSED WORK

A. Multichannel Assignment

In this section we present all the requirements that we would like to fulfill with our development, and we also enumerate the working assumptions that we have made.

Figure 2 presents a high-level architecture of the modified MobileNode with multi-channel support. Each node can have as many instances of the link layer, ARP, interface queue, MAC, network interface and channel entities as the number of interfaces it has. One can imagine that each instance actually represents a wireless network interface. Thus, this design scheme emulates the fact that our multi-channel multi-interface ad-hoc network implementation will not require any modification to existing IEEE 802.11 hardware.

As can be observed, most legacy operations of ns-2 are still preserved. Incoming traffic arrives through the corresponding channel and travel through the different components in ascending order then eventually merges to a single point at the address multiplexer. For outgoing traffic, the determination of selecting which interface to pass the data packets is to be handled by the routing...
agent. In other words, modifications will be required in implementing the routing agent to add the intelligence of selecting the appropriate interface.

In addition, the number of channels used in a single simulation could also be parameterized and nodes should be able to randomly connect to a subset of the defined wireless channels, thus giving a complete flexibility to the user. We understand that this level of flexibility, that needs to be accomplished from the scenario script, would be really important so as to evaluate different types of situations.

In addition, our intention is that the modified model could be used with any of the existing or new routing agents but it would also be nice being able to maintain the legacy behaviour of the simulator, so that already existing scripts would still be valid.

Requirements:
1. The number of channels in a particular scenario should be modifiable.
2. The number of interfaces per node is variable, and do not need to be the same for all nodes within a single scenario.
3. Each node within the same scenario could connect to a different number of channels (of the ones that had been previously defined).
4. Routing agents may take advantage of the modified model, but legacy operation of the simulator must be preserved, so as to ensure backwards compatibility.

B. Neighbourhood Load Routing

Problem description: In a WMN, the traffic may not be distributed evenly as some nodes are under light traffic load (transmitting or receiving a small amount of packets), while other nodes are under heavy traffic load (transmitting or receiving a large amount of packets). The problem of uneven traffic distribution caused overloaded traffic, is also defined as a load balancing problem.

To overcome centre node load balancing problem, the traffic load is supposed to be distributed evenly. In other words, the main objective that must be achieved is to keep the loads over different nodes comparable or even relatively equal. In an attempt to achieve this aim, our approach in this study consists of three stages. Firstly, the packets should travel on the lowest traffic load path instead of the shortest path. Secondly, a heavy loaded node should not be involved in forwarding packets. Thirdly, by reducing the interference of the network, the transmission waiting time of the packets decreases, thus, the overall traffic load of all nodes are reduced. Therefore, our proposal optimizes the traffic load distribution and reduces interference simultaneously.

In this section, we describe QoS-aware load balancing routing metric, namely, Neighbourhood Load Routing metric (NLR)[20]. It measures the average load of each neighbourhood and aim to bypass the busy neighbourhood instead of only bypassing the busy node by using LOAD COUNT routing metric.

Moreover, in a heavy loaded neighbourhood, extra traffic on one node generates interference to all its neighbourhood nodes and the transmission of packets in these nodes can be deferred, and then more packets are waiting in the IFQ.

To solve the above problem, NLR is used to check the average value of the neighbourhood load of a link which is:

\[ \text{NLR} = \sum_{i=1}^{k} \frac{\text{Load}_i^n}{b_i^n} \]

where \( n \) is the interference radius of neighbourhood in hop number, \( tr \) denotes the transmission range, \( d_{avg} \) is the average distance between two one-hop nodes. \( \text{Load}_i^n \) denotes the average load of a neighbourhood of node \( i \) with radius \( n \) hops and \( b_i \) is the average transmission rate of this neighbourhood. Hence, unlike the existing routing metrics, NLR considers three aspects in selecting a path, which are current IFQ length (packet size), neighbourhood interference, and neighbourhood bandwidth.

V. RESULT ANALYSIS

We use the NS-2 simulator [17] to evaluate the performance of AODV, WCETT and NLR. The simulation experiments aim to evaluate the performance of NLR in providing within the context of QoS provision of WMN. In our experiments, we compare HOP COUNT, WCETT and NLR in singlechannel WMN and also compare HOP COUNT and WCETT in multichannel WMN.

Two performance metrics are used which are average throughput, average end-to-end delay in the experiments. The average throughput calculates the capability of the network to accommodate traffic/messages. The average end-to-end delay is the average time a packet travels from a source to a destination.

In our simulation model the transmission range is 240m and simulation area extends up to 1000×1000m. Traffic source is FTP, and for each FTP session, the packet size is 1500 bytes and interval rate of 0.008 seconds. Following table shows the simulation
parameters required to analyze the performance of WMN supporting single-channel and multi-channel assignment.

<table>
<thead>
<tr>
<th>Simulation Time</th>
<th>80 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Area</td>
<td>1000×1000m</td>
</tr>
<tr>
<td>Propagation Model</td>
<td>Two-ray ground radio propagation</td>
</tr>
<tr>
<td>Antenna Model</td>
<td>Antenna/Omni Antenna</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>240m</td>
</tr>
<tr>
<td>Packet Size</td>
<td>1500 bytes</td>
</tr>
<tr>
<td>Interval Time</td>
<td>0.008 seconds</td>
</tr>
<tr>
<td>Address Type</td>
<td>Hierarchical</td>
</tr>
<tr>
<td>Channel Type</td>
<td>Channel/Wireless Channel</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>10, 20, 30, 40, 50</td>
</tr>
</tbody>
</table>

Table: Simulation Model

The following figures (namely Figure 3 and 4) shows the comparison of aodv-hop, wcett and NLR routing metrics by varying number of nodes from 10 to 50 in single-channel WMN using performance metrics average throughput and average end-to-end delay.

We can clearly see that NLR has distinguished performance in single-channel topology WMNs compared to other routing metrics.

WCETT ranks the second place in average throughput and AODV-hop achieves the lowest, whereas NLR is in the first place of examining of average throughput performance metric.

Aodv-hop and wcett suffer high communication time in delay while NLR scores the first place in average end-to-end delay.

To sum up the simulation results, we can clearly say that in single channel WMNs, NLR achieves high throughput than traditional hop-count and wcett routing metrics. The average end-to-delay is very high in aodv-hop and wcett, whereas NLR has low end-to-end delay.

**VI. CONCLUSION**

WMN needs multiple hop communication. Hence the nodes in WMN network especially, the Mesh Routers should have multiple channels. Here we have a routing protocol AODV and WCETT to support multiple channel interfaces for Mesh routers. In the simulation, we implement NLR in NS-2, and compare them with HOP COUNT and WCETT and the results show that NLR attains the lowest average end-to-end delay.
delay and highest average throughput in single-channel WMNs. This fully proves NLR outperforms other routing metrics in communication cost and capability to offer high standard QoS provision. As a future work, we plan to convert NLR to multichannel routing metric.

REFERENCES

A Separate Encryption and Decryption Service for Online BookStore in Cloud Computing

Ravindra Kumar Satuluri & Mr. D. Nagesh Babu
St. Ann’s College of Engineering and Technology, Chirala, India.
E-mail: satuluriravindra@gmail.com & dasarinagesh@yahoo.co.in

Abstract - Enterprises usually store data in internal storage and install firewalls to protect against intruders to access the data. They also standardize data access procedures to prevent insiders from accessing the information without permission. In cloud computing, the data will be stored in storage provided by service providers. Service providers must have a viable way to protect their clients’ data, especially to prevent the data from disclosure by unauthorized insiders. Storing the data in encrypted form is a common method of information privacy protection. If a cloud system is responsible for both tasks on storage and encryption/decryption of data, the system administrators may simultaneously obtain encrypted data and decryption keys. This allows them to access information without authorization and thus poses a risk to information privacy. This study proposes a business model for cloud computing based on the concept of separating the encryption and decryption service from the storage service. Furthermore, the party responsible for the data storage system must not store data in plaintext, and the party responsible for data encryption and decryption must delete all data upon the completion of encryption or decryption. A CRM (Customer Relationship Management) service is described in this paper as an example to illustrate the proposed business model. The exemplary service utilizes three cloud systems, including an encryption and decryption system, a storage system, and a CRM application system. One service provider operates the encryption and decryption system while other providers operate the storage and application systems, according to the core concept of the proposed business model. This paper further includes suggestions for a multi-party Service-Level Agreement (SLA) suitable for use in the proposed business model.

Keywords - cloud computing; service level agreements; encryption and decryption cloud service; data privacy protection.

I. INTRODUCTION

In recent years, cloud computing has become a hot topic in the global technology industry. The initiatives include Google’s research project for building an infrastructure to support research needs of top-tier American universities. Weiss noted that cloud computing services include several existing computing technologies [1], such as service-oriented utility computing [2], grid computing with large amount of computing resources [3], and that using data centers for data storage services.

Prior to the development of the concept of cloud computing, critical industrial data was stored internally on storage media, protected by security measures including firewalls to prevent external access to the data and including organizational regulations to prohibit unauthorized internal access. In the cloud computing environment, storage service providers must have in place data security practices to ensure that their clients’ data is safe from unauthorized access and disclosure. More importantly, the regulations and measures for preventing privileged users such as system administrators from unauthorized access must be rigorously established and implemented.

Service providers follow specific policies and practices to protect their users’ data, and these policies are usually stated in the service contract. Most current network application services have the same practice. For example, a Yahoo! web mail user must read the service contract online and show his consent to the service contract before he can use the web mail service. The content of the service contract covers definitions of service items, service scope, service change notification, scope of privacy protection, regulations on user data collection, use, sharing and release, and statements regarding user responsibilities. Showing the consent to the service contract is an essential step of the service application.

In a cloud computing environment, the service content offered by service providers can be adjusted according to the needs of the user. For example, the applicant can request different amounts of storage, transmission speeds, levels of data encryption and other services. In addition to defining the service items, the agreement normally also notes the time, quality and performance requirements provided with the service. Generally, these service agreements are referred to as Service Level Agreements (SLA) [4]. By signing an SLA, the user shows that he has understood and agreed
to the contents of the application service, and agrees with the provider’s data privacy and protection policies.

A common approach to protect user data is that user data is encrypted before it is stored. In a cloud computing environment, a user’s data can also be stored following additional encryption, but if the storage and encryption of a given user’s data is performed by the same service provider, the service provider’s internal staff (e.g., system administrators and authorized staff) can use their decryption keys and internal access privileges to access user data. From the user’s perspective, this could put his stored data at risk of unauthorized disclosure.

Creating user trust through the protection of user’s data content is the key to the widespread acceptance of cloud computing. This study proposes a business model for cloud computing based on the concept of using a separate encryption and decryption service. In the model, data storage and decryption of user data are provided separately by two distinct providers. In addition, those working with the data storage system will have no access to decrypted user data, and those working with user data encryption and decryption will delete all encrypted and decrypted user data after transferring the encrypted data to the system of the data storage service provider.

Under the business model proposed in this study, the data storage cloud system provider is authorized to store the user’s encrypted data, but does not have access to the Decryption Key. Thus, the storage system can only retrieve encrypted user data, but is unable to decrypt it. The cloud computing system responsible for encrypting user data has authority over all encryption keys required for data encryption but, given that the encryption provider does not store the user’s data, internal mismanagement of the decryption keys still poses no risk of unauthorized disclosure of the user’s data.

Given that encryption is an independent cloud computing service, a unique feature of the business model is that different services are provided by multiple operators. For example, the Encryption as a Service provider and the “Storage as a Service” provider cooperate to provide a Cloud Storage System with effective data protection. This study provides a draft SLA for this type of business model of combining multiple providers in a single service, which can establish the cooperation model between operators and the division of responsibility for the services they jointly provide to the user.

II. LITERATURE REVIEW

A. Origin and definition of cloud computing

The Internet began to grow rapidly in the 1990s and the increasingly sophisticated network infrastructure and increased bandwidth developed in recent years has dramatically enhanced the stability of various application services available to users through the Internet, thus marking the beginning of cloud computing network services. Cloud computing services use the Internet as a transmission medium and transform information technology resources into services for end-users, including software services, computing platform services, development platform services, and basic infrastructure leasing.

As a concept, cloud computing primary significance lies in allowing the end user to access computation resources through the Internet, as shown in Fig. 1. Some scholars find cloud computing similar to grid computing [3], but some also find similarities to utilities such as water and electrical power and refer to it as utility computing [2]. Because the use of resources can be independently adjusted, it is also sometimes referred to as autonomic computing [5].
B. Cloud computing business models

The hardware and architecture required for providing cloud computing environment services is similar to most computer hardware and software systems. The hardware in a modern personal computer (i.e., CPU, HDD, optical drive, etc.) performs basic functions such as performing calculations and storing data. The operating system (e.g., Windows XP) is the platform for the operations of the basic infrastructure, and text processing software such as MSWord and Excel are application services which run on the platform.

The architecture of cloud services can be divided into three levels: infrastructure, platform, and application software [7]. Application software constructs the user interface and presents the application system’s functions. Through the functions of the operations platform, the application can use the CPU and other hardware resources to execute calculations and access storage media and other equipment to store data.

Building a cloud computing application as a service requires infrastructure, platform and application software which can be obtained from a single provider or different service providers. If the revenue for cloud services primarily comes from charging for infrastructure, this business model can be referred to as Infrastructure as a Service (IaaS). If revenue comes primarily from charging for the platform, the business model can be referred to as Platform as a Service (PaaS). If revenue primarily comes from charging for applications or an operating system, the business model can be referred to as Software as a Service (SaaS).

Summarizing existing cloud services, Weihardt et al. proposed a holistic business model framework [8], as shown in Fig. 2.

Fig. 2 presents a hierarchical structure, with Platform as a Service as the value-added infrastructure service. The Application is built on the infrastructure and computing platform, and requires a specific user interface.

C. User data privacy concerns in a cloud computing environment

In a cloud computing environment, the equipment used for business operations can be leased from a single service provider along with the application, and the related business data can be stored on equipment provided by the same service provider. This type of arrangement can help a company save on hardware and software infrastructure costs, but storing the company’s data on the service provider’s equipment raises the possibility that important business information may be improperly disclosed to others[9].

Some researchers have suggested that user data stored on a service-provider’s equipment must be encrypted [10]. Encrypting data prior to storage is a common method of data protection, and service providers may be able to build firewalls to ensure that the decryption keys associated with encrypted user data are not disclosed to outsiders. However, if the decryption key and the encrypted data are held by the same service provider, it raises the possibility that high-level administrators within the service provider would have access to both the decryption key and the encrypted data, thus presenting a risk for the unauthorized disclosure of the user data.

D. Existing methods for protecting data stored in a cloud environment

Common methods for protecting user data include encryption prior to storage, user authentication procedures prior to storage or retrieval, and building secure channels for data transmission. These protection methods normally require cryptography algorithms and digital signature techniques, as explained below:

Common data encryption methods include symmetric and asymmetric cryptography algorithms. Symmetric cryptography is used in the U.S. Federal Information Processing Standard’s (FIPS) 46-3 Triple Data Encryption Algorithm (TDEA, also known as Triple-DES or 3DES) or 197 Advanced Encryption Standard (AES) and others. This type of encryption and decryption process uses a secret key. Asymmetric cryptography, on the other hand, uses two different keys, a “public key” for encryption, and a “private key” for decryption. Examples include RSA cryptography [11] and Elliptic Curve Cryptography (ECC) [12]. Generally speaking, symmetric cryptography is more efficient, and is suitable for encrypting large volumes of data. Asymmetric cryptography requires more computation time and is used for the decryption keys.
required for symmetric cryptography.

The use of passwords as an authentication process is more familiar to general users, but messages sent by the user are vulnerable to surreptitious recording by hackers who can then use the data in the message to log into the service as the user. In more advanced authentication systems, the system side will generate a random number to send the user a challenge message, requesting the user to transmit an encrypted response message in reply to the challenge message, thus authenticating that the user has the correct encryption key. Without this key, the user will not be allowed access. In the process of challenge and response the client’s encrypted key uses the client’s password to convert a derived value and. In this program, each communication between the client and server is unique, and a hacker using an old message would fail to access the system. In addition, the One-Time Password (OTP) authentication system differs from most peoples’ conception of a password[13]. Most people understand a password to be a password chosen by the user to be meaningful, and can be used again and again. The emphasis of OTP, however, is the single-use nature of the password.

After receiving authentication from the user, the system side must create a secure transmission channel to exchange information with the user. The Secure Sockets Layer (SSL) is a common method of building secure channels[14], primarily using RSA encryption to transmit the secret keys needed for the both sides to encrypt and decrypt data transmitted between them.

When using cryptographic technology to protect user data, the keys used for encryption and decryption of that data must be securely stored. In particular, cloud computing service providers must have specific methods for constraining internal system management personnel to prevent them from obtaining both encrypted data and their decryption keys – this is critical to protecting user data. Operator policies for protecting user data must be clearly laid out in the Service Level Agreement (SLA) and must explain how special privilege users are prevented from improperly accessing user data.

Kandukuri, Paturi and Rakshit offer six recommendations for SLA content[4], including (1) special privilege user data access must be controlled to prevent unauthorized storage or retrieval, (2) cloud computing services must comply with relevant laws, (3) user data must be properly stored and encrypted, (4) a reset mechanism must be provided in case of service disruption or system crash, (5) service must be sustainable and guaranteed against service discontinuation due to change or dissolution of the provider and (6) if cloud computing services are used for illegal purposes, the provider must be able to provide records to assist with investigations.

III. A BUSINESS MODEL FOR CLOUD COMPUTING BASED ON A SEPARATE ENCRYPTION AND DECRYPTION SERVICE

A. Core Concepts

This study proposes a Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service. The concept is based on separating the storage and encryption/decryption of user data, as shown in Fig. 3. In this business model, Encryption/Decryption as a Service and Storage as a Service (SaaS) are not provided by a single operator. In addition, the SaaS provider may not store unencrypted user data and, once the provider of Encryption/Decryption as a Service has finished encrypting the user data and handed it off to an application (e.g. a CRM system), the encryption/decryption system must delete all encrypted and decrypted user data.

Figure 3. Encryption/Decryption as an independent service

The concept of dividing authority is often applied in business management. For example, responsibility for a company’s finances is divided between the accountant and cashier. In business operations, the accountant is responsible for keeping accounts, while the cashier is responsible for making payments. By keeping these two functions separate, the company can prevent the accountant from falsifying accounts and embezzling corporate funds. Official documents frequently need to be stamped with two seals (i.e., the corporate seal and the legal representative’s seal), thus preventing a staff member from abusing his position to issue fake documents, and these seals are normally entrusted to two different people. These examples of the division of authority are designed to avoid a concentration of power which could raise operational risks.

In a cloud computing environment, the user normally uses cloud services with specific functions, e.g., Salesforce.com’s CRM service [15], SAP’s ERP services [16], etc. Data generated while using these services is then stored on storage facilities on the cloud service. This study emphasizes the addition of an independent encryption/decryption cloud service to this type of business model, with the result that two service
providers split responsibility for data storage and data encryption/decryption. To illustrate the concept of our proposed business model, Fig. 4 presents an example in which the user uses separate cloud services for CRM, storage and encryption/decryption. According to the user’s needs, CRM Cloud Services could be swapped for other function-specific application services (e.g., ERP Cloud Services, Account Software Cloud Services, Investment Portfolio Selection and Financial Operations Cloud Services).

Prior to the emergence of an emphasis on the independence of encryption/decryption services, CRM, ERP and other cloud services would simultaneously provide their users with storage services. This study emphasizes that Encryption/Decryption Cloud Services must be provided independently by a separate provider.

**B. Operating examples of the Encryption/Decryption as a Separate Cloud Service Business Model**

This section presents a CRM application service as an example of the new business model.

After the user logs into the CRM system, if the CRM Service System requires any client information, it will execute a Data Retrieval Program. When this data needs to be saved, it will execute a Data Storage Program. The Data Retrieval Program is illustrated in Fig. 5 and is explained below.

When a user wants to access the CRM Cloud Service, he must first execute the Login Program as shown in Step 1. This step can use current e-commerce or other services which have already securely verified the user’s registration, such as symmetric key-based challenge and reply login verification, or through a One-Time Password.

After the user’s login has been successfully verified, if the CRM Service System requires client information from the user, it sends a request for information to the Storage Service System, as shown in Step 2. In this step, the CRM Service System transmits the user ID to the Storage Service System where it searches for the user’s data. This data is encrypted so, once found, a request must be sent to the Encryption/Decryption Service System along with the user ID. Step 3 shows the Storage Service System executing the transmission of encrypted client data and the user ID to the Encryption/Decryption Service System.

Since the Encryption/Decryption Service System can serve multiple users and the encryption/decryption for each user’s data requires a different key, therefore each user’s unique ID and keys are stored together. Therefore, in Step 4, the Encryption/Decryption Service System uses the received user ID to index the user’s data decryption key, which is then used to decrypt the received data. Using the correct decryption key to decrypt the data is critical to restoring the data to its original state.

After the Encryption/Decryption Service System has decrypted the client’s data, in Step 5 the decrypted client data is provided to the CRM Service System which then displays the client data to the user in Step 6, completing the Data Retrieval Program. Prior to sending the decrypted client data, the Encryption/Decryption Service System and the CRM Service System can establish a secure data transmission channel (e.g., a Secure Sockets Layer connection) to securely transmit the decrypted client data. After the decrypted client data is sent, the Encryption/Decryption Service System is not allowed to retain the decrypted data and any unencrypted data must be deleted to prevent the encrypted data and the decryption key from being stored in the same system. This is a critical factor in ensuring the privacy of user data.

The above-mentioned Data Retrieval Program requires the collaboration of three different cloud service systems. Different methods of system collaboration are already supported by mature technologies, including two systems based on Universal Description Discovery and Integration (UDDI), Web Service Description Language (WSDL), and Simple Object Access Protocol (SOAP) to use Web Services or transmit Extensible Markup Language (XML) formatted...
data [17].

Next, we describe the Data Storage Program, as shown in Fig. 6. This program also involves the collaboration of three cloud service systems: CRM Service System, Encryption/Decryption Service System, and Storage Service System.

![Data Storage Diagram](image)

**Figure 6. Data storage diagram**

Step 1 of Fig. 6 shows the client sending a Data Storage Request to the CRM Service System which then initiates the Data Storage Program, requesting data encryption from the Encryption/Decryption Service System as shown in Step 2. In Step 2, the CRM Service System and the Encryption/Decryption Service System establish a secure data transfer channel to transmit the user ID and the data requiring storage from the CRM Service System to the Encryption/Decryption Service System.

As the encryption of data from different users requires different keys, in Step 3 the Encryption/Decryption Service System initiates data encryption, which involves using the received user ID to index the user’s encryption key which is then used to encrypt the received data.

Following this study’s emphasis on the principle of divided authority, once the client data is encrypted by the Encryption/Decryption Service System it must be transferred to the Storage Service System where the user ID and encrypted data are stored together.

Therefore, when the Encryption/Decryption Service System executes Step 4, it must transfer the user ID and encrypted client data to the Storage Service System. Step 5 shows the Storage Service System receiving the user ID paired with the data for storage. In this business model, the following the completion of Step 4 at the Encryption/Decryption Service System, all unencrypted and decrypted user data must be deleted.

Step 6, the final step of the Data Storage Program, transmits a Data Storage Complete message from the Storage Service System to the CRM Service System, at which point the CRM Service System may confirm that the client data has been stored. If it doesn’t receive a Data Storage Complete message, it can re-initiate the Data Storage Program or, after a given period of time, proceed with exceptional situation handling.

In the above example, the user’s goal in logging into the CRM Service System is possibly to maintain part of the client data, thus the system design must take data maintenance into consideration. Feasible design methods include matching the encrypted client data with the corresponding user ID and client ID, thus allowing for the indexing of the user ID to obtain the corresponding client data. Then the client ID can be used to index the client data the user wishes to maintain. Considering the massive amount of client data, search efficiency could be improved by combining the user ID and client ID to form a combined ID used for searching for a specific client’s data.

In the new business model, multiple cloud service operators jointly serve their clients through existing information technologies including various application systems such as ERP, accounting software, portfolio selection and financial operations which may require the user ID to be combined with other IDs for indexing stored or retrieved data. In addition, the foregoing description of the two systems can use Web Service related technology to achieve operational synergies and data exchange goals. These technologies can consider open international standards including the World Wide Web Consortium’s (W3C) published Web Service, UDDI, WSDL and SOAP standard documentation.

**IV. BENEFIT ANALYSIS AND DISCUSSION**

For cloud computing to spread, users must have a high level of trust in the methods by which service providers protect their data. This study proposes a Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service, emphasizing that authorization for the storage and encryption/decryption of user data must be vested with two different service providers. The privileges of Storage as Service provider include storing user data which has already been encrypted through an Encryption/Decryption Service System, but does not allow this service provider access to the Decryption Key or allow for the storage of decrypted data. Furthermore, the privileges of the Encryption/Decryption as Service provider includes management of the key required for the encryption/decryption of user data, but not the storage of decrypted or encrypted user data. In this new business model, user data in the Storage Service
System is all saved encrypted. Without the decryption key, there is no way for the service provider to access the user data. Within the Encryption/Decryption Service System there is no stored user data, thus eliminating the possibility that user data might be improperly disclosed.

After establishing “Independent Encryption/Decryption Services” in cloud computing environments, users of cloud computing services (e.g., CRM, ERP, etc.) will use the services of at least two cloud computing service providers, so agreements between these service providers are required to establish a model for cooperation and division of responsibilities in providing a common service to clients. This study provides a draft of a multi-signatory Service Level Agreement (SLA) in which the signatories can include cloud computing rental users, application service providers, encryption/decryption service providers, storage service providers, etc., with content including the rights and obligations between operators and also includes data security policies between each operator and clients.

REFERENCES


Facial Expression Recognition using DCT and MSE

Rohit D. Gawade & S. L. Nalbalwar
Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, Maharashtra, India
E-mail: rohitgawade7788@gmail.com, nalbalwar_sanjayan@yahoo.com

Abstract - The most expressive way humans display emotions is through facial expressions. In the past decade much progress has been made on face recognition, however computational facial expression analysis is still a challenging and attractive research topic in computer vision and intelligent human computer interaction. The automatic recognition of facial expressions has been an active research topic since the early nineties. In this paper we propose a new method to recognize facial expressions from images. First face portion is detected in face image, after that DCT (discrete cosine transform) is applied over face detected image to extract features and mean square error technique is used to recognize facial expressions in four categories i.e. happy, disgust, anger and neutral. Experimental results shows that the recognition rates are 100% and 83% for training and testing database respectively.

Keywords - Discrete Cosine Transform, Mean square Error

I. INTRODUCTION

Faces are accessible windows into the mechanisms which governs our emotional and social lives. About 70% of human communication is based on nonverbal communication such as facial expressions and body movements. Mehrabain pointed out that about 7% of human communication information is communicated by linguist language (verbal part), 38% by paralanguage (vocal part) and 55% by facial expressions. Therefore facial expressions provide the most information for emotion perception in face to face communication. Automatic facial expression analysis (AFEA) is gaining an interest in various application areas like lie detection, neurology, intelligent environments, clinical psychology, behavioral and cognitive sciences. It uses the facial signals as a new modality and causes the interaction between human and computer in a more robust and flexible way. A real time automatic surveillance system which detects human faces and facial expressions accurately can be installed at every airport around the world so that it can avoid the possible terrorist attack. In a real-time gaming application, facial expression recognition system can observe players facial expressions. A similar idea can be extended to application like driver observation system where a sleepy face of a driver can be traced by the camera and may indicate whether he or she is getting tired while driving. The systems mentioned above would be invaluable to help make a better world to live in. Those systems cannot be developed without robust algorithms to drive the software behind them.

Development of fully automatic facial expression recognition system is a challenging and complex topic in computer vision due to various factors like pose and illumination variations, different age, gender, ethnicity, facial hair, occlusion, head motions, lower intensity of expressions and other difficulties. Facial expression are generated by contraction and relaxation of facial muscles or by other psychological processes such as coloring of the skin, tears in eyes or sweat on the skin. Accuracy of facial expression recognition is mainly based on accurate extraction of facial feature components. From the survey, it is observed that various approaches have been used to detect facial features and classified as holistic and local feature based methods to extract facial features from images or video sequences of faces. These are geometry based, appearance based, template based and skin color segmentation based approaches. Two types of features can be extracted from the facial image: geometric features and appearance features. Geometric features measures the variations in shape, locations and distance of facial components like mouth, eyes, eyebrows, nose etc. in different expressions. In recent years, the research of developing automatic facial expression recognition systems has attracted a lot of attention.

II. BACKGROUND AND RELATED WORK

The initial studies on facial expressions started in the 17th century. John Bulwer wrote a book “Pathomyotomia” in which he gave a detailed note on various expressions and movement of head muscles in 1649. In 1667, at the Royal Academy of painting a
lecture was given by Le Brun which was later included in his book in 1734. In 1872 Darwin wrote a treatise that established the general principles of expression and the means of expressions in both humans and animals. Another important milestone in the study of facial expressions and human emotions is the work done by psychologist Paul Ekman and his colleagues since 1970s. Their work is of significant importance and has a large influence on the development of modern day automatic facial expression recognizers. The first step towards the automatic recognition of facial expressions was taken in 1978 by Suwa et al. Suwa and his colleagues presented a system for analyzing facial expressions from movie frames by using twenty tracking points[1]. Mase uses the means and variances of optical flow data at evenly divided small blocks [2]. Interframe motion of edges are extracted in the area of mouth, nose, eyes, and eyebrows by Yacoob and Davis to recognize facial expressions [3]. Essa and Pentland builds a dynamic parametric model by tracking facial motion over time [4]. Bartlett et al. use the combination of optical flow and principle components obtained from image differences[5]. Turk and Pentland represent face images by eigenfaces through linear principle component analysis[6]. Padgett and Cottrell use an approach similar to eigenfaces but with seven pixel blocks from feature regions (both eyes and mouth)[7]. Lanitis et al. use flexible models to analyze variations in shape and gray level appearance [8]. Rahardja et al. represented images in pyramid and use holistic representations with neural networks[9]. Cottrell and Metcalf used feed forward networks to extract principle components and based on this they used holistic representations to detect face expressions [10].

Tian et al. in 2001 used permanent features such as optical flow, Gabor wavelets and multistate models and transient features such as Canny edge detection algorithm to extract features and 2 artificial neural networks for facial expression classification. Bourel et al used local spatiotemporal vectors obtained from the EKLT tracker and modular classifiers with data fusion to classify facial expressions. In 2002, Paradas and Bonafonte extracted MPEG-4 Facial Animation Parameters using an improved Active Contour algorithm and motion estimation for feature extraction and hidden markov model for classification. In 2003 Cohen et al used a vector of extracted motion units (MUs) using PBVD tracker for feature extraction and Naive Bayes, Tree Augmented Naive Bayes, Stochastic Structure Search, Hidden Markov Model and Multi-level hidden Markov Model for facial expression classification. Bartlett et al., in 2003, used Gabor wavelets for feature extraction and SVM (Support Vector Machines) and AdaSVM (SVM with Adaboost) for classification. Michel and Kaliouby used Eucledian distance between neutral and peak for feature extraction and SVM for classification. In 2004, Pantic and Rothkrantz used frontal and profile facial points for feature extraction and rule based classifiers for facial expression classification. Pantic and Patras , in 2005 tracked a set of 20 facial fiducial points and used temporal rule to classify facial expressions. In 2006, Zheng et al. selected 34 landmark points on face image which are further converted into Labeled Graph using Gabor wavelet transform. Then a semantic expression vector built for each training face. Kernel canonical correlation analysis used to learn the correlation between LG vector and semantic vector. The correlation that is learnt is used to estimate semantic expression vector which is then use for classification. Anderson and McOwen used motion signatures obtained by tracking using spatial ratio template tracker and performing optical flow on the face using multichannel gradient model for feature extraction and SVM and multilayer perceptron neural network for classification. Aleksić and Katsaggelos used MPEG-4 facial animation parameters followed by PCA to reduce dimensionality and Hidden Markov Model and Multi Stream Hidden Markov model for facial expression classification. Pantic and Patras used Mid level parameters generated by tracking 15 facial points using particle filtering for feature extraction and Rule based classifiers for classification. In 2007, Sebe et al. used motion units generated from the PBVD tracker to extract features and Bayesian nets, SVMs and Decision trees, Used voting algorithms like bagging and Boosting for classification to improve results. Kotsia and Pitas used Geometric displacement of Candide nodes as features extracted and Multiclass SVM for facial expression classification. Wang and Yin used Topographic context (TC) expression descriptors for feature extraction and Quadric Discriminant Classifier, Linear Discriminant Classifiers, Support Vector Classifiers for expression classification. In 2008, Dornaika and Davoine used Candide face model to track features and expressions are recognized using stochastic approach. Kotsia et al. used Gabor features, Distributed Nonnegative Matrix Factorization algorithm and Geometric displacements vectors extracted using Candide tracker for feature extraction and Multiclass SVM and Multi-Layer perceptron for facial expression classification [1].

III. DISCRETE COSINE TRANSFORM

The Discrete Cosine Transform represents an image as a sum of sinusoids of varying magnitudes and frequencies. The DCT has a property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT. Like other transform, the discrete cosine transform attempts to decorrelate the image data. The most common DCT definition of a one dimensional sequence of length N is

\begin{equation}
\text{DCT}\{x[n] \} = \frac{2}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] \cos \left( \frac{\pi}{N} (n+\frac{1}{2}) \right) k,
\end{equation}

where \( x[n] \) is the input sequence and \( \{DCT\{x[n]\}\} \) is the output sequence. The DCT is a linear, orthogonal transformation, which transforms a signal from its time domain to its frequency domain. The DCT is widely used in image and video compression, as well as in other fields such as audio processing and communication systems.
Facial Expression Recognition using DCT and MSE

\[ C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos \left( \frac{\pi(2x+1)u}{2N} \right) \]  \hspace{1cm} (1)

for \( u=0,1,2,\ldots,N-1 \). Similarly the inverse transformation is defined as

\[ f(x) = \sum_{u=0}^{N-1} \alpha(u)c(u) \cos \left( \frac{\pi(2x+1)u}{2N} \right) \]  \hspace{1cm} (2)

for \( x=0,1,2,\ldots,N-1 \). In both equations 1 and 2 \( \alpha(u) \) is defined as

\[ \alpha(u) = \begin{cases} \frac{1}{\sqrt{N}} & \text{for } u=0 \\ \frac{2}{\sqrt{N}} & \text{for } u \neq 0 \end{cases} \]  \hspace{1cm} (3)

It is clear from 1 that for \( u=0 \),

\[ c(u = 0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \]  \hspace{1cm} (4)

Thus the first transform coefficient is the average value of the sample sequence.

The two dimensional DCT is a direct extension of the one dimensional case and is given by

\[ c(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left( \frac{\pi(2x+1)u}{2N} \right) \cos \left( \frac{\pi(2y+1)v}{2N} \right) \]  \hspace{1cm} (5)

For \( u,v=0,1,2,\ldots,N-1 \) and \( \alpha(u) \) and \( \alpha(v) \) are defined in 3. The inverse transform is defined as

\[ f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v)c(u, v) \cos \left( \frac{\pi(2x+1)u}{2N} \right) \cos \left( \frac{\pi(2y+1)v}{2N} \right) \]  \hspace{1cm} (6)

for \( x,y=0,1,2,\ldots,N-1 \). The two dimensional basis function can be generated by multiplying the horizontally oriented 1-D basis functions with vertically oriented set of same functions. On applying DCT, the input image will get decomposed into a set of basis images. For highly correlated data, cosine transform show excellent energy compaction. Most of the energy will be represented by a few transform coefficients [11].

IV. METHODOLOGY

Let us have \( M \) images in training database. Fig. 1 shows the detailed methodology used to develop facial expression recognition system. It consists of mainly two phases training and testing.

A. Training Phase:
- All training expression image samples are resized to common size.
- Face portion of each training face image is detected.
- Features are extracted from each face detected train image using discrete cosine transform technique.
- A model of extracted features of all train images is formed which consists of DCT coefficients of all train images.

B. Testing Phase:
- Test expression image sample whose expression is detected is resized to common size as that of training image.
- Face portion of test face image is detected
- Features are extracted from face detected test image using discrete cosine transform technique.
- Extracted features which consists of DCT coefficients are compared with the features of train images which are stored in a model formed while training.
- Mean square errors between each test image features and all train image features are calculated.
- Minimum value of MSE is detected to recognize the expression of test face image under observation.
EXPERIMENTAL RESULTS.

Our database consists of 41 train images and 31 test images having different expressions. Confusion matrix of both train and test images for detecting the expression are shown in tables below. Expressions in columns corresponds to expressions recognized by our system and those in rows corresponds to expressions determined by human observer.

TABLE I. CONFUSION MATRIX OBTAINED FOR TRAINING DATABASE

<table>
<thead>
<tr>
<th></th>
<th>Happy</th>
<th>Disgust</th>
<th>Anger</th>
<th>Neutral</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>Disgust</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>Anger</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Neutral</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Mean recognition rate: 100%

TABLE II. CONFUSION MATRIX OBTAINED FOR TESTING DATABASE

<table>
<thead>
<tr>
<th></th>
<th>Happy</th>
<th>Disgust</th>
<th>Anger</th>
<th>Neutral</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Disgust</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>Anger</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Neutral</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Mean recognition rate: 83%

REFERENCES


---

International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati
Region Classification Based Near Lossless Image Compression

S. Abinav & V. K. Govindan
Dept. of Computer Science and Engineering, National Institute of Technology Calicut
Kerala-673601, India
E-mail: cegabi@gmail.com, vkg@nitc.ac.in

Abstract – Image compression allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. High quality of compressed image is required in many fields such as remote sensing and medicine. We have proposed a region classification based near lossless image compression approach. An average compression percentage of 53.54 is obtained when tested with ten images.

Keywords - Image Compression; region classification; Lempel-Ziv-Welch.

I. INTRODUCTION

Lots of images are being generated by many devices which occupy a huge amount of space. In many fields, like medicine, military affairs and scientific research, we need good image quality and low storage consumption. RLE (Run Length Encoding), Huffman, LZW (Lempel-Ziv-Welch), arithmetic encoding are all well known lossless compression methods. In the case of lossless compression we are able to get back the exact image without pixel loss. However, they cannot provide very high compression ratio. In the case ROI (Region Of Interest), regions which are important are compressed using lossless methods and the remaining are compressed using lossy.

Lempel Ziv Welch (LZW) [1] is a well known lossless compression algorithm. In this method characters are grouped till we get a sequence of characters which is not in the dictionary. We output the corresponding code from the dictionary without the recent input character and the sequence with the input is added into the dictionary. Steps in the algorithm are as follows:

1. Dictionary is initialised with single characters.
2. We find the largest string match of the input with the dictionary.
3. Output the code for the matched string from dictionary and remove the string from input.
4. Push the matched string and the next input character to the dictionary.
5. Go to Step 2.

Huffman coding [2] is a variable length coding for lossless compression. In this, the characters or symbols are encoded using a code table. The code table is constructed based on the frequencies or probabilities of character symbols in the input to be encoded. Huffman coding uses prefix free codes, that is, the bit string representing a particular character is never a prefix of the code representing another character.

Frequently occurring characters will require lesser number of bits than the once which are less frequent. Figure 1 shows a Huffman tree. With each symbol we associate a frequency, symbols with least frequency of occurrence are combined together to form a binary tree. Frequency of the root will be the sum of the frequencies of its child. We repeat the same procedure till all encoding symbols are in the binary tree.

II. LITERATURE SURVEY

Some of the related work in this topic are: region of interest coding by Men and Heng [3], the work by Chien-Wen et al. [4] on near lossless wavelet compression technique for satellite images, lossless image compression technique employing optimal super-spatial prediction of structural components proposed by Xiwen and Zhihai [5], adaptive wavelet transform approach by Guang-quan and Li-zhi [6], row by row classification and LZW encoding approach by Xie et al. [7], hyper-spectral image compression based on quadtree partitioning proposed by Zhang et al. [8], the approach by Luo et al. [9], and the DCT image compression based on genetic algorithm proposed by Chen et al. [10]. These are briefly reviewed in the following.
Men and Heng-Ming [3] proposed region of interest coding for image compression. This paper presents a new method for region of interest (ROI) coding based on the embedded block coding with optimized truncation (EBCOV paradigm). The proposed method reduces the priority of the less important region or background of an image, allowing the user to quickly view the ROI with higher quality without receiving the entire image. This method substantially saves the transmission time, storage space, and computational cost of image compression. The unique characteristics of the new method lie in the combination of good ROI compression performance and low algorithm complexity. The major drawbacks of these methods are that these methods elevate the number of bit-planes to be encoded/decoded due to the coefficient shift, which increases the algorithm complexity and execution time. Another important work is that by Chien-Wen et al. [4] for near lossless satellite image compression employing wavelet technique. In the proposed method, when each ROI is reconstructed, it can reconstruct near lossless images with less bit rate than the recommendation of CCSDS does. Benefited from run-length coding and specific residue image bit-plane compensation, the proposed algorithm can obtain higher quality satellite images at similar bit rate or lower bit rate at the similar image quality. This work can be further improved by combining other binary compression techniques and the extension of this work may offer a VLSI or a DSP implementation of the proposed algorithm. Xiwen and Zhihai [5] proposed an improved approach for image compression employing super-spatial prediction of structural components. The super-spatial prediction approach attempts to optimally predict the structural components within the whole image domain. The authors considered only lossless image compression based on a minimum spanning tree, developed an optimum prediction scheme for structural components. One major drawback of the proposed method is its computational complexity. The major complexity lies in the construction of the minimum spanning tree for optimum prediction. Guang-quan and Li-zhi [6] proposed a new image compression via adaptive wavelet transform. The authors proposed a new adaptive DWT via image texture. Partitioning the image based on local content, predicting the texture direction of local image, keeping the orthogonal property, and applying directional wavelet filtering the authors achieved improved performance by lifting structure with in-place operation.

Xie et al. [7] proposed a near lossless compression approach for images employing row by row classification of pixels and LZW encoding. The algorithm takes advantage of the effects of the distribution of pixels on compression ratio. The pixels are classified row by row firstly, and the classification result is recorded in a mask image. After that the image data is decomposed into two sequences, and the mask image is hidden in them. Finally, the sequences are encoded using LZW algorithm. The result showed that the proposed algorithm reduced the bit rate of compression ratio than LZW, RLE, and Huffman encoding; and the PSNR is large. The proposed algorithm is also easy to implement. Although some noise is produced in the compressed images, they are not significant. Zhang et al. [8] proposed Compression of hyper-spectral images based on quadtree partitioning. The paper analyzes the characteristic features of hyper-spectral images and presents a compression of hyper-spectral images based on quadtree partitioning. Quadtree partition is used to get the mean image of the whole image and the significant correlation of the image can be decorrelated by subtracting the mean image from original image. The difference image is compressed by DCT and encoded with arithmetic code. Luo et al. [9] proposed an easy image compression method and its realization based on Matlab. The algorithm of the JPEG image compression based on DCT was discussed firstly. A new JPEG compression method based on pre-filtering before compressing was proposed, which can cope with the problem of impairing dependency of noisy digital image. Chen et al. [10] proposed Discrete Cosine Transform image compression based on genetic algorithm. The approach used threshold selection, crossover and random mutation to achieve high compression. The optimal threshold selection is done based on genetic algorithm.

The approach proposed in the present work is a near lossless compression technique employing region classification, LZW and Huffman encoding, which is an extension of the work proposed by Xie et al. [7] for near lossless compression of images employing row by row classification of pixels and LZW encoding.
III. PROPOSED METHOD

Pre-processing steps have been done for a given image in Matlab. The image is read in a matrix, and region based classification is done to group foreground and background values. The block diagram representation of the proposed Algorithm is as shown in Figure 2.

3.1. Region based classification

Each square region of same size in the image matrix is processed individually. The regions are thresholded using an optimum threshold based on histogram shape. Threshold for each region is found using Otsu thresholding given in [11]. The algorithm assumes that the region to be thresholded has a bimodal histogram. The algorithm is reproduced as shown below:

To following symbols are required to be defined:

\( \omega_i \) : probability of \( i^{th} \) class, \( i=1,2 \)

\( \sigma_i \) : variance of \( i^{th} \) class

\( t \) : Threshold of intensity value

\( \mu_i \) : Mean of \( i^{th} \) class

\( \omega_i(t) \): probability of class \( i \) separated by threshold \( t \) and variance \( \sigma_i^2(t) \).

All pixels with values less than or equal to threshold are assumed to be in class \( \omega_1 \), and the rest in class \( \omega_2 \).

So, the probability of class 1 and 2 are computed as:

\[
\omega_1(t) = \sum_j p(j)
\]

\[
\omega_2(t) = \sum_{j=t+1}^H p(j)
\]

where \( H \) is the highest intensity in the image block.

Mean values of class 1 and class 2 are computed as:

\[
\mu_1(t) = \sum_j p(j)x(j)
\]

\[
\mu_2(t) = \sum_{j=t+1}^H p(j)x(j)
\]

The intraclass variance is defined as the weighted sum:

\[
\sigma_b^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)
\]

Otsu’s algorithm determine the optimum threshold \( t \) by searching for a \( t \) that minimizes the above weighted sum, \( \sigma_b^2(t) \).

**Algorithm 1:**

**Input**: each block or region of the image  
**Output**: Optimum threshold of the block

**Step 1**: Compute histogram, \( p \) of the block, and the probabilities of each intensity level.

**Step 2**: Set up initial \( \omega_i(0) \) and \( \mu_i(0) \)

**Step 3**: Step through all possible thresholds \( t = 1 \) to \( H \)

Maximum intensity, \( H \)

\( \Rightarrow \) Update \( \omega_i(t) \) and \( \mu_i(t) \)

\( \Rightarrow \) Compute \( \sigma_b^2(t) \)

**Step 4**: Desired threshold corresponds to the minimum \( \sigma_b^2(t) \)

We find the optimum threshold for each square region using otsu thresholding and again apply otsu thresholding as a whole on the threshold value’s which we get from each region. Now we have a final threshold value to convert the image to a mask image. The image is thresholded by using the rule: Pixels having intensity less than or equal to the threshold \( t \) are given value zero and pixels having intensity higher than threshold \( t \) are...
given values one. So we get a mask image of the original image which is nothing but a binary image. This mask image will be useful for classifying the original image.

3.2. Decomposition of original image

Based on the mask image we classify the pixels in the original image into two sequences - class1 and class2. If the mask image pixel value is one then the corresponding pixel values in original image is put into class1 and if the pixel value is zero then the corresponding pixel values in original image is put in class2. So we get two sequences class1 and class2 with high gray values and low gray values respectively.

3.3. Hiding the mask image

The mask is needed for decoding, but it occupies extra space. In order to reduce the bits per pixel of the compressed image, the mask is hidden into the least significant bits of class1 and class2. We hide the first element of mask image into the first element of class1 and second element of mask image into second element of class1 and so on till all elements of class1 are exhausted after which we hide the mask image into elements of class2 as given in [7]. When a binary value b is hidden into an unsigned byte variable B, the result will be:

\[ B = B \& 254, \text{ if } b \text{ is zero} \]
\[ B = B | 1, \text{ if } b \text{ is one} \]

3.4. LZW and Huffman encoding

Further these two classes have to be compressed separately using LZW algorithm. It is found that the output after compression by LZW algorithm contains more redundant terms. Huffman encoding has the ability to remove redundant values by assigning less number of bits to frequently occurring values. Hence, we get a better compression efficiency.

The LZW algorithm [1] is given in algorithm listing below:

**Algorithm 2:**

set \( I = \text{NIL} \)
loop
read a character \( J \)
if \( IJ \) exists in the dictionary
\( I = IJ \)
else
output the code for \( I \)
add \( IJ \) to the dictionary
\( I = J \)
endloop

Huffman encoding [2] steps are given in algorithm listing below:

**Algorithm 3:**

Step1: Create two nodes with the character and its probability
Step2: Choose two nodes having least probability.
Step3: Make a new node as the parent of nodes having least probability.
Step4: The new node is given a value which is sum of probabilities of child node.
Step5: Continue until only one parentless node is left.

Compressed data is Huffman decoded followed by LZW decompression to retrieve class1 and class2. The mask image is extracted from class1 and class2 and based on the mask image each pixel value is obtained either from class1 or class2. This is just the reverse process of compression to get the image back.

<table>
<thead>
<tr>
<th>Image name</th>
<th>Image1</th>
<th>Image2</th>
<th>Image3</th>
<th>Image4</th>
<th>Image5</th>
<th>Image6</th>
<th>Image7</th>
<th>Image8</th>
<th>Image9</th>
<th>Image10</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>original size (58 x 30 x 8)</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
<td>460800</td>
</tr>
<tr>
<td>Compressed size using existing method [7]</td>
<td>316330</td>
<td>436895</td>
<td>350430</td>
<td>387198</td>
<td>344428</td>
<td>472830</td>
<td>230058</td>
<td>228300</td>
<td>139334</td>
<td>219875</td>
<td>312197.9</td>
</tr>
<tr>
<td>Compressed size using proposed method</td>
<td>239915</td>
<td>336028</td>
<td>224056</td>
<td>289082</td>
<td>247045</td>
<td>247516</td>
<td>175974</td>
<td>173208</td>
<td>109070</td>
<td>167819</td>
<td>213451.3</td>
</tr>
<tr>
<td>Percentage compression using existing method [7]</td>
<td>31.35</td>
<td>5.18</td>
<td>23.00</td>
<td>15.97</td>
<td>25.25</td>
<td>-2.61</td>
<td>49.04</td>
<td>51.43</td>
<td>69.76</td>
<td>52.28</td>
<td>32.28</td>
</tr>
<tr>
<td>Percentage compression using proposed method</td>
<td>49.08</td>
<td>31.41</td>
<td>51.27</td>
<td>35.08</td>
<td>54.93</td>
<td>46.28</td>
<td>61.81</td>
<td>62.41</td>
<td>78.08</td>
<td>63.58</td>
<td>53.54</td>
</tr>
</tbody>
</table>
IV. EXPERIMENTAL RESULT

The performance of the proposed approach is compared with the existing row by row classification [7] approach using a database of ten images. The results are tabulated as in Table 1. With the existing method, an average compression of 32.28 percentage was observed. The proposed approach provides 53.54 percent as the average compression, which is better than the compression performance of original method.

V. CONCLUSION

High quality of compressed image has many applications. However, compression performances of lossless techniques are normally very low. We have proposed a novel near lossless image compression algorithm using region based classification followed by LZW and Huffman coding. The algorithm is easy to implement and has higher compression performance than some of the existing methods which use just Huffman coding. However there is a very small loss of image bits due to the hiding of mask image into the least significant bits of the image data, which is negligible. The proposed approach provides a average compression of 53.54 percent.

REFERENCES

Counting Bloom Filter Architecture in VLSI Network Systems

Nagamalli. A¹ & Kedareswararao. M²

¹Department of E.C.E, A.I.E.T, Narsipatnam, Visakhapatnam, Andhra Pradesh, India
²Department of E C E, A.I.E.T, Narsipatnam, J. N. T. U. K, Kakinada, India
E-mail : mallika.arasavalli@gmail.com¹, Kedhar_mutyala@yahoo.com²

Abstract – the Counting Bloom Filter (CBF) is useful for real time applications where the time and space efficiency is the main consideration in performing a set membership tests. The CBF estimates whether an element is present in a large array or not by allowing false positives and by not permitting false negatives. In this paper CBF architecture is analysed and has been implemented. There are two approaches of CBF, SRAM based approach using up/down counters and the LCBF using up/down LFSR unit. In this paper the LCBF architecture discussed and analyzed. In the latest VLSI technology it is easy to fabricate memories that hold a few million bits of data and addresses. But in the recent embedded memory technologies rather than mapping of addresses of 5000 bits of data using hashing functions we can concise into single contiguous memory.

Keywords - Bloom filter, Hashing mechanism, SRAM, LCBF, false positives, false negatives, look-ups, VLSI network systems

I. INTRODUCTION

In 1970 BURTON HOWARD BLOOM proposed a elegant probabilistic data structure for testing set-membership tests, in the process of finding automatic hyphenation using an English dictionary. As there is limited core memory RAM that time, therefore the entire dictionary of the English language could not be held there. It is somewhat paradoxical to have to wait for an era of ever increasing core memory for Bloom filters to broadly take hold. Fig 1 shows the Bloom filter architecture.

Fig. 1 : The Bloom Filter Architecture

Simply stated, a Bloom filters achieves space efficiency, less than O (mn) space, by allowing for a small probability of false positives, but no false negatives, to the set-membership question. Bloom Filter is a space efficient probabilistic data structure which is used for testing the presence of an element.

II. PROPERTIES AND TYPES OF THE BLOOM FILTER

A. PROPERTIES

Two useful properties of the Bloom filter for access control are:

1. The union of two Bloom filters of the same size and using the same hash functions can be obtained by bitwise ORing the two filters.
2. The intersection of two Bloom filters of the same size and using the same hash functions can be obtained by bitwise ANDing the two filters.
B. TYPES

Bloomier filters

In the case of "Bloomer filters[6]", a false positive is defined as returning a result when the key is not in the map. The map will never return the wrong value for a key that is in the map.

Stable Bloom filters

The Stable Bloom filter introduces false negatives, which do not appear in traditional bloom filters.

Scalable Bloom filters

The technique is based on sequences of standard bloom filters with increasing capacity and tighter false positive probabilities, so as to ensure that a maximum false positive probability can be set beforehand, regardless of the number of elements to be inserted.

Attenuated Bloom filters

The attenuated filter of level i indicates which services can be found on nodes that are i-hops away from the current node.

Counting Bloom Filter (CBF)

An increasing number of architectural techniques rely on hardware counting bloom filters (CBFs) to improve upon the power, latency and complexity of various key processor structures. CBF dynamically bypasses the conventional mechanism as frequently as possible. Accordingly, the benefits obtained through the use of a CBF depend on how frequently it can be utilized and on the CBF's energy and latency characteristics. The more tests are serviced by the CBF alone and the lower the power and latency of the CBF, the higher the benefits. Architectural techniques and application behavior determine how many tests can be serviced by the CBF.

In this paper CBF architecture is analysed and applications are discussed for network systems.

III. CBF ARCHITECTURE

Counting Bloom Filter (CBF) consists of a small element addresses array that is connected to the array through hashing mechanism, i.e. multiple addresses mapped to single array [1], as shown in fig. 3

Fig. 4 shows CBF with single memory vector for a typical 5000 bits.

Fig. 4: Bloom filter with single memory vector

CBF has three operations increment count (inc), decrement count (dec) and test if count is zero (probing) as shown in fig [3]. CBF is characterized by its number of entries and width of the count per entry. CBF can be accessed faster than any other data structure and needs very less energy when compared to that of accessing a large set of data, most of the membership tests are serviced by the CBF.

In the latest VLSI technology it is easy to fabricate memories that hold a few million bits of data and addresses. But in the recent embedded memory technologies rather than mapping of addresses of 5000 bits of data using hashing functions we can concise in to a single contiguous memory [3].

IV. CBF ALGORITHM

We want to test for membership in a set S = of n elements. The universe U of elements is typically very large. Let h1, h2, h3, hk be k independent hash functions with range

Bloom filter consists of a bit array of m – bits with all set to 0. There must be k- hash functions defined each one hashes some set element to one of the m-array positions in a uniform random distribution format.

Adding an element: feed to each of the k hash functions to get k array positions. Set the bits at all these positions to 1.

Query for an element: To find whether the element is in it or not feed it to each of the k hash functions to get k array positions. If any of the bits at these positions are 0, the element is definitely not in the set; if it were, then all the bits would have been set to 1 when it was inserted. If all are 1, then either the element is in the set, or the bits have by chance been set to 1 during the insertion of other elements, this is called as false positives.

False positives are acceptable but false negatives are not accepted, because removing an element from
the simple bloom is not possible, once the element is removed from the composite filter re-adding is also impractical[4]. This problem can overcome in counting bloom filter architectures.

Basic steps followed by bloom filters with initial assumed state as empty,

1. Empty bloom filter is a bit array of m ‘0’ bits.
2. Introduce ‘k’ hash functions, each maps key value to one of m array positions.
3. Insert element by feeding it to each hash function, to obtain k array positions. Set these bits to ‘1’.
4. Query an element by re-feeding it in to each hash function, and checking corresponding bit positions. If all bits are set 1, then either the element is in the filter or a false positive.
5. If bit positions of hashes of an element contains ‘0’, then that element is definitely not in the filter; no false negatives.

This can be represented in an array as shown in below fig [5]

![Fig.5: Working of bloom filters with empty, insertion and query operations](image)

V. ESTIMATION OF FALSE POSITIVE PROBABILITY (\( \hat{P} \))

The false positive probability \( \hat{P} \) is a function of \( \mathbf{n} \) number of elements in the filter of size \( \mathbf{m} \). Assuming an optimal number of hash functions, \( k = \frac{\mathbf{m}}{\mathbf{n} \ln 2} \).

Assume that a hash function selects each array position with equal probability. If \( m \) is the number of bits in the array, the probability that a certain bit is not set to one by a certain hash function during the insertion of an element is then 

\[
\frac{m-n}{m} \approx 1 - \frac{1}{e} \approx 0.632.
\]

The probability that it is not set by any of the hash functions is.

\[
\approx e^{-\frac{n}{m}}.
\]

If we have inserted \( n \) elements, the probability that a certain bit is still 0 is

\[
\approx e^{-\frac{n}{m}}.
\]

The probability that it is 1 is therefore

\[
\approx 1 - e^{-\frac{n}{m}}.
\]

Now test membership of an element that is not in the set. Each of the \( k \) array positions computed by the hash functions is 1 with a probability as above. The probability of all of them being 1, which would cause the algorithm to erroneously claim that the element is in the set, is often given as

\[
\left(1 - \left(1 - \frac{1}{m}\right)^k\right) \approx \left(1 - e^{-\frac{k}{m}}\right)^k.
\]

This is not strictly correct as it assumes independence for the probabilities of each bit being set. However, assuming it is a close approximation we have that the probability of false positives decreases as \( m \) (the number of bits in the array) increases, and increases as \( n \) (the number of inserted elements) increases. For a given \( m \) and \( n \), the value of \( k \) (the number of hash functions) that minimizes the probability is

\[
\frac{m}{n} \ln 2 \approx 0.7\frac{m}{n}.
\]

This gives the false positive probability of

\[
2^{-k} \approx 0.6185^{\frac{m}{n}}.
\]

The required number of bits \( m \), given \( n \) (the number of inserted elements) and a desired false positive probability \( p \) (and assuming the optimal value of \( k \) is used) can be computed by substituting the optimal value of \( k \) in the probability expression above:
\[ p = \left(1 - e^{-\frac{m \ln 2}{n \ln n}}\right)^{m/n \ln 2} \]  
(1)

And can be simplified to

\[ \ln p = -\frac{m}{n} \left(\ln 2\right)^2 \]  
(2)

This results in:

\[ m = -\frac{n \ln p}{(\ln 2)^2} \]  
(3)

This means that for a given false positive probability \( p \), the length of a Bloom filter \( m \) is proportionate to the number of elements being filtered \( n \). While the above formula is asymptotic (i.e. applicable as \( m, n \to \infty \)), the agreement with finite values of \( m, n \) is also quite good; the false positive probability for a finite bloom filter with \( m \) bits, \( n \) elements, and \( k \) hash functions is at most

\[ \left(1 - e^{-\frac{(n+0.5)/(m-1)}}\right)^k \]  
(4)

So we can use the asymptotic formula if we pay a penalty for at most half an extra element and at most one fewer bit [1].

**VI. COUNTING BLOOM FILTER IMPLEMENTATION**

CBF implementation can be handled in two methods

1. SCBF – SRAM BASED CBF
2. LCBF– LOWPOWER CBF

In earlier work SRAM based CBF (SCBF) was implemented, where up/down counters were used in place of up/down LFSR unit in each partition.

In LCBF architecture input is an address given to a pre decoder. Through global decoder it is transferred to local decoder and gated clock circuitry, where the clock is generated according to the input address and local decoded data.

Basic cells consist of DWL based decoder structure, up/down LFSR unit and a zero detector with local and global multiplexer units. For a 3 level decoder DWL STRUCTURE can be represented as shown below in fig. 7

![Fig.7 DWL architecture using a three level decoder [4]](image)

The critical path for a typical decoder implemented using the DWL architecture as shown in fig. 8

![Fig.8 Critical path for a 3 level decoder](image)

Decoders encompasses the circuits from the address input to the word line as shown in fig. 6

Design of each decoder consists of combinatorial modes:

DR CMOS – delayed reset logic uses delayed version of one of the inputs to conditionally reset the gate

SR CMOS – self resetting logic uses output to reset the gate

![Fig.9 Schematic of fast- low power three-level decoder structure](image)

---

International Conference on Computer Science and Information Technology, ISBN : 978-93-81693-86-5, 10th June, 2012-Tirupati
DRCMOS techniques will have the least logical effort and hence the lowest delay.

Next block in internal structure of LCBF is up/down LFSR unit with Galois implementation. LFSR will produce a pseudorandom sequence of maximal length $2^{n-1}$ states, where $n$ is the number of stages. The sequence will then repeat from the initial state for as long as the LFSR is clocked[5]. An example of a 3-bit LFSR and truth table are shown in fig. 10[7].

Thus changing the tap points and setting the preset value many pseudo random sequences will be generated; following truth table is an example.

Table (1) Truth table for 3-bit LFSR making tap 0 as 1

<table>
<thead>
<tr>
<th>LFSR stage</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>Value</th>
<th>Binary equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESET-0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PRESET-1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Output of each LFSR given to a zero detector which is selected according to the reset logic and the corresponding result transferred through local multiplexer, global multiplexer, as a result output of global multiplexer combinely gives set membership test result as “is-zero” as shown in fig. 6.

VII. CONCLUSION

Main advantage of using LFSR in the CBF architecture is switching noise can be reduced as there is no “rollover” unlike a binary counter. By and then various types of bloom filters were studied. CBF architecture is analyzed to obtain probability of minimum false positive.

Thus by using 3 level decoder structure with “DRCMOS logic” delay can be reduced as the decoders designed with lowest delay and critical path delay also reduced for a 3 level decoder of LCBF.

Thus LCBF estimates and speeds the set membership test for an element with minimum false positives, and reduced path delay structures in DWL decoders.

For efficient tradeoff between delay and energy in a large range RAM structures in VLSI can be designed by the simple mechanism of varying the sizes of the word driver inputs and the total logical effort can be reduced significantly by skewing the gates in the inner structures of DWL decoders of CBF.

A technique where Constant time computation of the algorithm along with the scalability is needed, such as applications of network intrusion detections, which require real time processing then LCBF is a best method of implementing the hardware architecture.

ACKNOWLEDGMENT

We acknowledge with thanks our respective college colleagues who have given support for successful presentation of this paper.

We also thank Sri G.TirumalaRao, Professor, Department of E.C.E, G.V.P. College of Engineering, for his valuable suggestions in completing this paper.

REFERENCES


Hash Based Intrusion Detection and Forensic Analysis of Tampered Database

Jayshree T. Agale & Shefali .P.Sonavane
Walchand College of Engineering, Sangli
E-mail : Jayu.agale@gmail.com, shefali.sonavane@gmail.com

Abstract – Secure data storage is a need of many clients like financial institutes, public services and government agencies. By using cryptographically strong hash functions, tampering detection can be done. This paper detects database intrusion using extensive hash calculations and comparisons. The detected data is further forensically analyzed in temporal and spatial domain using tiled bitmap algorithm. Advantage of this algorithm is the detection of multiple corruption events in same tile (time interval).

Keywords - Integrity, consistency, tampering, forensic analysis

I. INTRODUCTION

There is a necessity to have secure data house where different types of clients can store their data. Database of organizations and all entities are very crucial as it shows the identity of organization which they wants to be secret for outsiders. There should be need to control access the database even by insiders. There are chances of tampering the database either by insiders or outsiders to serve their purposes. Tampering is illegal alteration of data or modifying data in any way and that activity is known as corruption event. Corruption event is any event that corrupts the data and compromises the database [1]. It could be due to an adversary, including an auditor or an employee or even an unknown bug in the software or a hardware failure.

The question may arise in our mind that why such tampering events are taking place. There are various reasons in which tampering can be done either by insiders or outsiders. Let’s discuss one example wherein any unsatisfied student who gets ‘C’ grade in mathematics but actually need ‘B’ grade in that subject can attempt to change the student’s database unless he gets an access to the student’s database containing grades of student. This is one example in which the intruder is an outsider. Let’s have other example in which intruder is employee in private organization. Suppose an employee have to fill minimum sales requirement for that particular year to improve his performance. If he couldn’t fill his requirements for that year then he would highly tempted to change date of transaction to make it appear that sales requirement for that year is fulfilled but in actual he couldn’t do that. This is other example of tampering in which intruder is insider. Recent experience has shown that sometimes the (human) auditors themselves cannot be trusted. (A recent security survey states that the source of cyber security breaches appear fairly evenly split between those originating on the outside and those originating on the inside [2].). Previously proposed a mechanism within a database management system (DBMS), based on cryptographically strong one-way hash functions, which prevent an intruder, including an auditor or an employee or even an unknown bug within the DBMS itself, from silently corrupting the audit log [3].

In order to detect tampering there is a need to perform hashing on the data generated after every transaction and then validating it against audit log database [1]. In validation phase, it sends the hash value, which is computed over the entire database to the notarization service, which is external digital service indicating whether that value matches one previously computed. If two hash values does not match then tampering is occurred over there. After tampering detection, there comes role of forensic analysis. Previously three algorithms have been developed like monochromatic, RGB, polychromatic for forensic analysis [4]. They were having their own disadvantages like they were not able to detect multiple corruption events within same time period and there were unnecessary computation of additional hash chains. So tiled bitmap algorithm comes in picture which is refinement of previous algorithms removing previous limitations [3].
This paper uses efficient way of hashing data. In notarization service, document is digitally notarized so there are no chances at all to tamper that document so originality of the master data is preserved for further comparison.

II. PROBLEM DEFINITION

Input: Test data set that has to be secured

Processing:

- Hashing of the data generated by every transaction and then comparing computed hash value with hash value generated from notarized document for the purpose of tamper detection.

- Forensic analysis of detected tamper using tiled bitmap algorithm

Output: it finds 1] Tampering detection

2] Tampering analysis in terms of temporal and spatial estimation.

III. PROPOSED WORK

The major components of the proposed system are DBMS, notarization service, Validator and forensic analyzer. The secure data house or data care centre is maintained storing data of different clients like online book shop, banking and hospital management. So database is created in starting phase using MySQL as back end.

Clients are accessing this developed application through web. Data entries inserted by clients get stored to server site where application is actually running. At that server site implementation is carried out in following manner.

Before clients start manipulating the data, one trusted party at every client’s site uploads their digital document or master data on server’s machine. All attributes of every row along with digital document are concatenated and given as input to hash algorithm as shown in fig.1, which then outputs hash key. Same processing is carried out at server on master data or original data and computed hash value is compared with previously generated hash value and match is checked out. If the values don’t match, tampering is has been occurred.

The master data is digitally notarized as stated before. Sample pdf file is used as certificate where the owner’s identity is provided with some watermarking or any authentication key that is other security major provided in paper.
Every transaction is hashed when it is committed and then it is linked to the previous transaction to create hash chain $C_0$. Hash chain is created by linking hash keys of transaction in sequence. Validation period or interval ($I_v$) i.e. the time period between two successive validation is defined for e.g. 16 and after every validation period, system runs Validator which rehashes every transaction and then compares value with previously notarized hash value.

If $C_0$ gives false result (for mismatch) then partial hash chains are computed. This computation of partial hash chains are for the further forensic analysis. Collection of partial hash chain generated in validation period is known as tile (T). Tile means time period provided for each validation. Partial hash chains within one tile (T) are denoted by $C_0(T), C_1(T), C_2(T).... C_{log(I_v)}(T)$ with $C_i(T)$ denoting the $i^{th}$ hash chain of tile starting at time ‘T’. Number of partial hash chains in each tile is equal to $\log (I_v/R_s)$ so last chain is given as $C_{log(I_v)}(T)[1]$. Where $R_s$ is spatial detection resolution. The finest spatial granularity of the corrupted data would be an explicit attribute of a tuple or a particular time stamp attribute. By considering few hours and skipping next hours, partial hash chain is formed. Hash chain linking is explained in detail in paper [3].

```
// input: $\tau_{FY,F}$ is the time of first validation failure
// $I_v$ is the validation interval
// k is used for the creation of $C_{I,R}$
// $R_s$ is the spatial detection resolution
// output: $C_{set}$, an array of binary numbers

function TiledBitmap($\tau_{FY,F}, I_v, k, R_s$)
1: $l \leftarrow 0$  // the target
2: $C_{set} \leftarrow C_{tile} \leftarrow \emptyset$
3: $r \leftarrow 1$
4: while $r < \tau_{FY,F}$ do
5:   if $\neg$ val_check($\tau_{FY,F}$) then
6:     $n \leftarrow \log(I_v/R_s)$
7:     for $i \leftarrow n$ to 1
8:       $t \leftarrow t + 2^n \cdot$ val_check($C_i(r)$)
9:     $C_{temp} \leftarrow$ candidateSet($l, n, k$)
10:    for each $r \in C_{temp}$
11:      $y \leftarrow$ remnumber($r$, $v$, $R_s$)
12:     $C_{set} \leftarrow C_{set} \cup \{g\}$
13: $r \leftarrow r + I_v$
14: return $C_{set}$
```

Fig.3 Tiled bitmap algorithm

As given in algorithm fig.3, $I_v$ is number of hours between two successive validation termed as validation interval. Helper function val_check is used which takes partial hash chains as input parameter and then generates Boolean result of that chain. On line 4, algorithm iterates through different tile and checks longest partial chain $C_0$. If $C_0$ gives Boolean result as true then algorithm switches to next tile. If it evaluates to false then algorithm moves through rest of the partial hash chains in same tile and concatenates result of each hash chain. After concatenation target binary number is formed. Then candidateSet function is called at line 9 to compute all candidate sets from target binary number. Candidate set means possible location of tampering of tampering. It may be in term of hours.

The different functions used like candidateSet, createRightmost, generate and funkySort functions are given with algorithm in [1].

The user-specified parameter $k$ discussed in [1]. By default its value is taken as 2.

CandidateSet function is called to compute all the candidate set elements from the target number. In candidate set generation target binary string $t=0000$ and $000$ are treated as two different strings.

IV. PERFORMANCE ANALYSIS

Data house for different clients is maintained as stated before. For online book shop client, the paper have drawn some results.

After creating database for clients like online book shop, some of the book entries are filled up and whenever wrong data gets inserted or illegal updation or deletion is done it gives message like intrusion detected. It provides information along with user name, at which date and at what time.

Case 1: If data is entered successfully without any forgery for book id BL999

Master Query is--------
R GandhiMahadevDesaiNavaJeevan200501012/12/20101
2/12/2013
Log Query is----------
R GandhiMahadevDesaiNavaJeevan200501012/12/20101
2/12/2013

Key1: 06a192b7ead73108aff6739197e6ea5e
Key2: 06a192b7ead73108aff6739197e6ea5e

Final Result is: An Intrusion is not detected.

Case2: If data is entered with forgery for book id BL999
Master Query is-
GandhiMahadevDesaiNavaJeevan200501012/12/201012/12/2013

Log Query is-
GandhiMahadevDesaiNavaJeevan2005001012/12/201012/12/2013

Key1: 06a192b7ead73108aff6739197e6ea5e

Key2: b1e58d80de9a84459f750e9ad810c4b3

Final Result is: An Intrusion is happened while INSERTING, so the CANDIDATESET IS BL999 from User suraj on 13-04-2012 at 4:25:48 PM......

Graph1. Efficiency of hash

As shown in graph avalanche effect is always high no matter how many bits are changed in input.
An Empirical Analysis of Single Model Test Prioritization Strategies for Event Driven Software

P. Dileep Kumar Reddy & A. Ananda Rao
Department of CSE, JNTUA College of Engineering, Ananthapur, India
E-mail : dileep_kumar_reddy@yahoo.co.in, akepogu@gmail.com

Abstract – Software testing can be stated as the process of validating and verifying the software program or application or product. Web and Event-driven applications (EDS) is a class of applications that is quickly becoming ubiquitous. In Event Driven Software (EDS) the number of input events leads to large number of states and require large number of test cases. So, common testing strategies for both Graphical User Interface (GUI) and web applications because both have similarities if combined into a single abstract model, conventional testing strategies do not apply in some cases. The results of the study will be promising as many of the prioritization criteria that used helps in improving the rate of fault detection. The tests applied in the study are for the web applications come from real user-sessions, whereas the GUI test cases were automatically generated without influence from users and are user friendly in nature. Both are particularly challenging to test because users can invoke many different sequences of events that affect application behavior. Hence here we propose a novel model to rank the test cases based on their prioritization.

Keywords - event driven software (EDS), test suite prioritization, web application testing, GUI testing.

I. INTRODUCTION

Event driven software is user friendly and easy to interact with application. GUI and web applications are the basic examples to state that Event-driven software (EDS) [1] can change state based on incoming events. Earlier researches showed that existing conventional testing techniques do not apply to either GUIs or web application [6] [3]. Because number of input events lead to large number of states, and require large number of test cases. So, there are similarities between GUI and Web application. Hence, Common testing strategies for GUI and web application have to be followed. Combining both into abstract single model will give good result.

This work provides first single model that is enough to analyze GUI and web application combindly. GUI based application is front-end to software’s Underlying back-end code. Changes in states usually reflect to changes in GUI widget also. Graphical user interface object has a fixed set of properties and hierarchical in nature. Graphical front-end accepts user generated input and gives graphical output. A web application consists of pages that are accessible by users through a browser and it is transmitted to other users over a network. Web application includes both client side and server side languages. Testing the web program is largely manual task but capture-replay tools help in less time consumption. Applying the test prioritization strategies after combining into a single model is already discussed in the paper.

The further extension of evaluating and generalizing the model by various testing activities such as test generation prioritization criteria improve the rate of fault detection depends uniformly on overall cost. Analysis of costs of faults leads to hybrid prioritization experiments. Hybrid prioritization gives better fault detection mechanism. Empirical analysis model helps to verify properties required to improve the single-model strategies. The ultimate goal is proper testing of event driven software.

II. RELATED WORK

The new testing techniques for GUI which has a fixed set of properties and hierarchical in nature and web based applications in which pages are accessed by user through browser and transmit over network and these techniques are discussed below. The GUI and web applications are two basic examples of EDS. Session driven applications are used in web application and Event flow application are used in GUI.

2.1 Session Driven Applications

A web site can be differentiated from a web application based on the “ability of a user to affect the state of the business logic on the server” [4]. In other words, requests made of a web application go beyond navigational requests, including some form of data that
needs further decomposition and analysis to be served. Figure 1 shows how a simple web application operates. A user (client) sends a request through a web browser. The web server responds by delivering content to the client. This content generally takes the form of some markup language (e.g., HTML) that is later interpreted by the browser to render a web page. For example, if a request consists of just a URL (Uniform Resource Locator – a web site address), the server may just fetch a static web page. Other requests are more complicated and require further infrastructure. For example, in an e-commerce site, a request might include not just a URL, but also data provided by the user. Users provide data primarily through forms consisting of input fields (textboxes, checkboxes, selection lists) rendered in a web page. This information is translated into a set of name-value pairs (input fields’ names and their values) and becomes part of the request. Although the web server receives this request, further elements are needed to process it. First, a group of scripts and perhaps an application server may parse the request, query a database server to retrieve information about the item, and then employ further formatting scripts to generate HTML code on the fly to address the user request. This newly generated page, generated at run time and depending on user’s input, is called a dynamic web page [2] [6].

2.2 Event Flow Applications

Daily Automated Regression Tester is used to test the GUI. It analyzes each widget in each of the window of the project. It computes the total number of possible Smoketest cases and the test designer specifies the number of test cases that should be executed. For GUI smoke testing, a tester has to produce test cases that satisfy requirements [7] [8].

- The smoke test cases should be generated and executed quickly i.e in one night...The test cases should provide adequate coverage of the GUI’s functionality. As is the case with smoke test cases of conventional software, the goal is to raise a “something is wrong here” alarm by checking that GUI events and event-interaction execute correctly.
- As the GUI is modified, many of the test cases should remain usable. The goal here is to design testcases that are robust, in that a majority of them remain unaffected by changes to the GUI.
- The smoke test should be divisible into parts that can be run on different machines [10].

The most common tools used to test GUIs are capture/replay tools like WinRunner that offer little automation particularly for making check cases. There are tries to develop state-machine models to automate some aspects of GUI testing; e.g., test-case generation and regression testing. GUI testing, during this paper, is outlined as exercising the complete application by generating solely GUI inputs with the intent of finding failures that manifest themselves through GUI widgets.

III. EMPIRICAL ANALYSES AND HYBRIDIZING THE TEST PRIORITIZATION APPROACH

Empirical analysis model helps to verify properties required to improve the single model strategies using hybrid prioritization. Empirical analysis model helps to improve the rate of fault detection. The function takes as input a set of test cases to be ordered, and returns a sequence that is ordered by the prioritization criterion. Combined model is used in the work which has prioritization criteria and also prioritization function has been considered [1].

3.1 User Session based test prioritization

User session based techniques one limiting factor in the use of white box web application testing techniques such as Ricca and Tonella’s is the cost of finding inputs that exercise the system as desired. Selection of such inputs is slow and must be accomplished manually [4]. User-session based techniques can help with this problem by transparently collecting user interactions and transforming them into test cases. The techniques capture and store the clients’ requests in the form of URLs and name-value pairs, and then apply strategies to these to generate test cases.

Our first technique, User Session to Test Case transformation (USTCT), transforms each individual user session into a test case. Given m user sessions, $U_1, U_2, U_3, ..., U_m$, with user session $U_i$ consisting of n requests $r_1, r_2, r_3, ..., r_n$, where each $r_j$ consists of url[ name - value]*, the test case corresponding to $U_i$ is generated by formatting each of the requests, from $r_1$ to $r_n$, into an http request that can be sent to a web server. The resulting test suite contains m test cases, one for each user session. (For simplicity, we define a user session as beginning when a request from a new IP address reaches the server and ending when the user leaves the web site or the session times out.) Our second technique user interactive test case transformation (UITCT) user-session based technique, UITCT, generates new user sessions based on the pool of collected data, creating test cases that contain requests belonging to different users. UITCT is meant to expose error conditions caused by the use of sometimes
conflicting data provided by different users. UITCT generates a test case as follows:

- Randomly select unused session \( U_a \) from session pool;
- Copy requests \( r_1 \) through \( r_i \), where ‘i’ is a random number greater than 1 but smaller than \( n \), into the test case;
- Randomly select session \( U_b \), where \( b \neq a \), and search for any \( r_j \) with the same URL as \( r_i \), and if an equivalent request is not found, select another session \( U_b \);
- Add all the requests from \( U_a \) after \( r_j \) to the test case;
- Mark \( U_a \) “used”, and repeat the process until no more unused sessions are available in the pool.

Our approach is meant to be hybridized even to test event flow based applications, applied either in the beta testing phase to generate a baseline test suite based on interactions under beta version, or during subsequent maintenance to enhance a test suite that was originally generated by a more traditional method. Further, the approach can help testers monitor and improve test suite quality as the web application evolves, and as its usage proceeds beyond the bounds anticipated in earlier releases and earlier testing [1].

3.2 GUI Test case Prioritization using Event flow

Where a test case is a sequence of events that a user invokes through the GUI, early coverage of all unique events in a test suite, and early event interaction coverage between windows (i.e., select tests that contain combinations of events invoked from different windows which have not been covered in previously selected tests). In half of these experiments, event interaction-based prioritization results in the fastest fault detection rate. The two applications that cover a larger percentage of interactions in their test suites (64.58% and 99.34% respectively) benefit from prioritization by interaction coverage. The applications that cover a smaller percentage of interactions in their test suites (46.34% and 50.75% respectively) do not benefit from prioritization by interaction coverage.

This test process is in the sequence of

- First it checks how many components are on the window.
- Then it checks whether they are initialized or not.
- Then checks whether they are added on frame or not.

- Then checks each component have action listener or not.
- Then checks label of each component to check different component having same name or not
- After finding out the faults, we are displaying all those faults with simplified messages.
- Whether developer forgot to set frame visible true or not. We are checking whether he/she set the frame visibility mode true or not.

3.3 Hybrid Model

To advance the Hybrid Model, we conduct aboriginal analysis, how GUI and web applications operate. In GUI models, the programmer accouterments an action listener that acknowledges the user’s adumbration, which is some implementation-dependent activity should occur. When the user performs an event, e.g., clicks a button, chooses a card item, an activity accident occurs.

For instance, in web applications, there may be assorted user interactions one individual window in which users set ethics for ambit afore any advice is in fact beatific to the web server (e.g., a POST or GET request). To advance consistency in our analogue for both GUI and web applications, we arrange the appellation activity to be the after set of allures interactions on a individual window afore affective to a new window [9].

IV. EXPERIMENTS AND RESULTS ANALYSIS

The performance of the proposed hybrid test prioritization model in short can refer as HTPM was tested on event call tree generated from a sample java swing application and a simulated web application session log with 22 sessions. With the results, it is evident that test case ranking based on their priority was achieved by minimizing the 97.5% of redundancy in test case selection while retaining the scalability in execution. Figure 1and 2 indicates the advantage of HTPM over single model for Test case prioritization (SM-TCP) [5] in terms of test case ranking by their priority.

![Fig. 1: A line chart representation of Memory utilization by single model and hybrid model.](image-url)
V. CONCLUSION

Here in this paper we proposed a Hybrid Test case Prioritization Model alleged HTCPM in adverse to Previous works those treat stand-alone GUI and web-based applications as abstracted areas of research. However, these types of applications accept abounding similarities that acquiesce to actualize a Hybrid Model for testing such event apprenticed systems. This archetypal may advance approaching analysis to add broadly focus on stand-alone GUI and web based applications instead of acclamation them as break topics. Other advisers can use our accepted archetypal to administer testing techniques added broadly. Within the ambience of this model, we advance and empirically appraise several prioritization criteria. Our empiric abstraction evaluates the prioritization criteria. Our adeptness to advance prioritization belief for two types of event-driven software indicates the account of our Hybrid Model for the issue of test prioritization. Our after-effects are able as abounding of the prioritization belief that we use advance the amount of accountability apprehension over accidental acclimation of analysis cases. We accurate the archetypal through the appliance of test suit prioritization by applying several prioritization criteria.

REFERENCES


Compressing Group Movement Patterns for Tracking Moving Objects Efficiently

Mahammad Habibulla Shaik, A.V.S.Sudhakara Rao & P.Harini
Department of CSE, St.Ann’s College of Engineering & Technology, Chirala, India
E-mail : mohhabibulla@gmail.com, ande_sudhakar@yahoo.co.in, hpogadadanda@yahoo.co.in

Abstract – In this paper, we first propose an efficient distributed mining algorithm to jointly identify a group of moving objects and discover their movement patterns in object tracking applications, we propose a compression algorithm, called 2P2D, which exploits the obtained group movement patterns to reduce the amount of delivered data. The compression algorithm includes a sequence merge and an entropy reduction phases. We formulate a Hit Item Replacement (HIR) problem and propose a Replace algorithm that obtains the optimal solution. Moreover, we devise three replacement rules and derive the maximum compression ratio. Our experiment results show that the proposed compression algorithm effectively reduces the amount of Delivered data and enhances compressibility and, by extension, reduces the energy consumption expense for data transmission in WSNs. We are using these algorithms to find out the performances on the bases on time complexity and space complexity by using visualization technique.

Keywords - Data Compression, Distributed Clustering, Object Tracking.

I. INTRODUCTION

RECENT advances in location-acquisition technologies, such as global positioning systems (GPSs) and wireless sensor networks (WSNs) have fostered many novel applications like object tracking, environmental monitoring, and location-dependent service. These applications generate a large amount of location data, and thus, lead to transmission and storage challenges, especially in resource constrained environments like WSNs. To reduce the data volume, various algorithms have been proposed for data Compression and data aggregation [1], [2], [3], [4], [5], [6]. However, sequential patterns 1) consider the characteristics of all objects, 2) lack information about a frequent pattern’s significance regarding individual trajectories, and 3) carry no time information between consecutive items, which make them unsuitable for location prediction and similarity. In addition, most of the above works are centralized algorithms [9], [10]. We thus define the problem of compressing the location data of a group of moving objects as the group data compression problem. Therefore, in this paper, we first introduce our distributed mining algorithm to approach the moving object clustering problem and discover group movement patterns. Our distributed mining algorithm comprises a Group Movement Pattern Mining (GMPMine) algorithm. Different from previous compression techniques that remove redundancy of data according to the regularity within the data, we devise a novel two-phase and 2D algorithm, called 2P2D, which utilizes the discovered group movement patterns shared by the transmitting node and the receiving node to compress data. Specifically, the 2P2D algorithm comprises a sequence merge and an entropy reduction phases.

II. EXISTING SYSTEM

In the Existing System Discovering the group movement patterns is more difficult than finding the patterns of a single object or all objects, because we need to jointly identify a group of objects and discover their aggregated group movement patterns.

The constrained resource of WSNs should also be considered in approaching the moving object clustering problem. However, few of existing approaches consider these issues simultaneously. On the one hand, the temporal-and-spatial correlations in the movements of moving objects are modeled as sequential patterns in data mining to discover the frequent movement patterns.

However, sequential patterns
1) Consider the characteristics of all objects,
2) Lack information about a frequent pattern’s significance regarding individual trajectories,
3) Carry no time information between consecutive items, which make them unsuitable for location prediction and similarity comparison.

A. Disadvantages of Existing System:

On the other hand, previous works, such as measure the similarity among these entire trajectory sequences to group moving objects. Since objects may be close together in some types of terrain, such as gorges, and widely distributed in less rugged areas, their group relationships are distinct in some areas and vague in others. Thus, approaches that perform clustering among entire trajectories may not be able to identify the local group relationships.

In addition, most of the above works are centralized algorithms which need to collect all data to a server before processing it causes, unnecessary and redundant data may be delivered, leading to much more power consumption because data transmission needs more power than data processing in Wireless Sensor Networks (WSNs).

III. PROPOSED SYSTEM

In this paper, we first introduce our distributed mining algorithm to approach the moving object clustering problem and discover group movement patterns. Our distributed mining algorithm comprises a Group Movement Pattern Mining (GMPMine) algorithm. Different from previous compression techniques that remove redundancy of data according to the regularity within the data, we devise a novel two-phase and 2D algorithm, called 2P2D, which utilizes the discovered group movement patterns shared by the transmitting node and the receiving node to compress data. Specifically, the 2P2D algorithm comprises a sequence merge and an entropy reduction phases. In the sequence merge phase, we propose a Merge algorithm to merge and compress the location data of a group of objects. In the entropy reduction phase, we formulate a Hit Item Replacement (HIR) problem to minimize the entropy of the merged data and propose a Replace algorithm to obtain the optimal solution. We formulate the HIR problem to minimize the entropy of location data and explore the Shannon’s theorem to solve the HIR problem.

A. Mining of group Movement Patterns

To tackle the moving object clustering problem, we propose a distributed mining algorithm, which comprises the GMPMine and CE algorithms. First, the GMPMine algorithm uses a PST to generate an object’s significant movement patterns and computes the similarity of two objects by using simp to derive the local grouping results. The merits of simp include its accuracy and efficiency: First, simp considers the significances of each movement pattern regarding to individual objects so that it achieves better accuracy in similarity comparison. For a PST can be used to predict a pattern’s occurrence probability, which is viewed as the significance of the pattern regarding the PST, simp thus includes movement patterns’ predicted occurrence probabilities to provide fine-grained similarity comparison. Second, simp can offer seamless and efficient comparison for the applications with evolving and evolutionary similarity relationships. This is because simp can compare the similarity of two data streams only on the changed mature nodes of emission trees [11], instead of all nodes.

B. The Group Movement Pattern Mining (GMPMine) Algorithm

To provide better discrimination accuracy, we propose a new similarity measure simp to compare the similarity of two objects. For each of their significant movement patterns, the new similarity measure considers not merely two probability distributions but also two weight factors, i.e., the significance of the pattern regarding to each PST. The similarity score simp of o_i and o_j based on their respective PSTs, T_i and T_j, is defined as follows:

\[
\text{simp}(o_i, o_j) = -\log \sum_{s \in S} \left( \sum_{\sigma \in T_i(s)} P_i^T(\sigma) - \sum_{\sigma \in T_j(s)} P_j^T(\sigma) \right)^2,
\]

Where e S denotes the union of significant patterns (node strings) on the two trees. The similarity score simp includes the distance associated with a pattern s, defined as

\[
d(s) = \sqrt{\sum_{\sigma \in T_i} (P_i^T(\sigma) - P_j^T(\sigma))^2} = \sqrt{\sum_{\sigma \in T_j} (P_j^T(\sigma) - P_i^T(\sigma))^2},
\]

where d(s) is the Euclidean distance associated with a pattern s over T_i and T_j.

For a pattern S \in T, P^T(S) is a significant value because the occurrence probability of s is higher than the minimal support P_{min}. Note that the distance between two PST’s is normalized by its maximal value, i.e.,

2L_{max} + \sqrt{2}. We take the negative log of the distance between two PSTs as the similarity score such that a larger value of the similarity score implies a stronger similar relationship, and vice versa. With the definition of similarity score, two objects are similar to each other if their score is above a specified similarity threshold.
The GMPMine algorithm includes four steps. First, we extract the movement patterns from the location sequences by learning a PST for each object. Second, our algorithm constructs an undirected, unweighted similarity graph where similar objects share an edge between each other. We model the density of group relationship by the connectivity of a sub graph, which is also defined as the minimal cut of the sub graph. When the ratio of the connectivity to the size of the sub group is one of the most energy expensive tasks in WSNs, data compression is utilized to reduce the amount of delivered data [1], [2], [3], [4], [5], [6]. The algorithm includes the sequence merge phase and the entropy reduction phase to compress location sequences vertically and horizontally. In the sequence merge phase, we propose the Merge algorithm to compress the location sequences of a group of moving objects. The Merge algorithm avoids redundant sending of their locations, and thus, reduces the overall sequence length. It combines the sequences of a group of moving objects by 1) trimming multiple identical symbols at the same time interval into a single symbol or 2) choosing a qualified symbol to represent them when a tolerance of loss of accuracy is specified by the application. Therefore, the algorithm trims and prunes more items when the group size is larger and the group relationships are more distinct. In the entropy reduction phase, we propose the Replace algorithm that utilizes the group movement patterns as the prediction model to further compress the merged sequence. The Replace algorithm guarantees the reduction of a sequence’s entropy, and consequently, improves compressibility without loss of information. To reduce the entropy of a location sequence, based on which the Replace algorithm reduces the entropy efficiently.

In sequence merge Phase we concentrate on the problem of compressing multiple similar sequences of a group of moving objects.

In the entropy reduction phase, we propose the Replace algorithm to minimize the entropy of the merged sequence obtained in the sequence merge phase. Since data with lower entropy require fewer bits for storage and transmission [11], we replace some items to reduce the entropy without loss of information. The object movement patterns discovered by our distributed mining algorithm enable us to find the replaceable items and facilitate the selection of items in our compression algorithm. In this section, we first introduce and define the HIR problem, and then, explore the properties of Shannon’s entropy to solve the HIR problem. We extend the concentration property for entropy reduction and discuss the benefits of replacing multiple symbols simultaneously. We derive three replacement rules for the HIR problem and prove that the entropy of the obtained solution is minimized.

A. The Replace Algorithm

Based on the observations described in the previous section, we propose the Replace algorithm that leverages the three replacement rules to obtain the optimal solution for the HIR problem. Our algorithm examines the predictable symbols on their statistics, which include the number of items and the number of predictable items of each predictable symbol.

The algorithm first replaces the qualified symbols according to the accumulation rule. Afterward, since the concentration rule and the multiple symbol rule are related to \(n(i)\), which is increased after every
restitution, the algorithm iteratively replaces the qualified symbols according to the two rules until all qualified symbols are replaced. The algorithm thereby replaces qualified symbols and reduces the entropy toward the optimum gradually. Compared with the brute force method that enumerates all possible intermediate sequences for the optimum in exponential complexity, the Replace algorithm that leverages the derived rules to obtain the optimal solution in \( O(L) \) time is more scalable and efficient. We prove that the Replace algorithm guarantees to reduce the entropy monotonically and obtains the optimal solution of the HIR problem.

V. EXPERIMENT & ANALYSIS

We compare our batch-based approach with an online approach for the overall system performance evaluation and study the impact of the group size (n), as well as the group dispersion radius (GDR), the batch period (D), and the error bound of accuracy (eb). We also compare our Replace algorithm with Huffman encoding technique to show its effectiveness. Since there is no related work that finds real location data of group moving objects, we generate the location data, i.e., the coordinates \((x, y)\), with the Reference Point Group Mobility Model [16] for a group of objects moving in a two-layer tracking network with 256 nodes. A location-dependent mobility model [17] is used to simulate the roaming behavior of a group leader; the other member objects are followers that are uniformly distributed within a specified group dispersion radius (GDR) of the leader, where the GDR is the maximal hop count between followers and the leader. We utilize the GDR to control the dispersion degree of the objects. Smaller GDR implies stronger group relationships, i.e., objects are closer together. The speed of each object is 1 node per time unit, and the tracking interval is 0.5 time unit. In addition, the starting point and the furthest point reached by the leader object are randomly selected, and the movement range of a group of objects is the Euclidean distance between the two points. Note that we take the group leader as a virtual object to control the roaming behavior of a group of moving objects and exclude it in calculating the data traffic.

Next, our compression algorithm utilizes the group relationships to reduce the data size. The amount of data per object decreases as the group size increases. Compared with carrying the location data for a single object by an individual packet, our batch-based approach aggregates and compresses packets of multiple objects such that the amount of data decreases as the group size increases. Moreover, our algorithm achieves high compression ratio in two ways. First, while more sequences that are similar or sequences that are more similar are compressed simultaneously, the Merge algorithm achieves higher compression ratio. Second, with the regularity in the movements of a group of objects, the Replace algorithm minimizes the entropy which also leads to higher compression ratio. Note that we use the GDR to control the group dispersion range of the input workload. The leader object’s movement path together with the GDR sets up a spacious area where the member objects are randomly distributed.

The amount of data decreases as the batch period increases. Since more packets are aggregated and more data are compressed for a longer batch period, our batch-based approach reduces both the data volume of packet headers and the location data.

VI CONCLUSIONS

In this work, we exploit the characteristics of group movements to discover the information about groups of moving objects in tracking applications. We propose a distributed mining algorithm, which consists of a local GMPMine algorithm and a CE algorithm, to discover group movement patterns. With the discovered information, we devise the 2P2D algorithm, which comprises a sequence merge phase and an entropy reduction phase. In the sequence merge phase, we propose the Merge algorithm to merge the location sequences of a group of moving objects with the goal of reducing the overall sequence length. In the entropy reduction phase, we formulate the HIR problem and propose a Replace algorithm to tackle the HIR problem. In addition, we devise and prove three replacement rules, with which the Replace algorithm obtains the optimal solution of HIR efficiently. Our experimental results show that the proposed compression algorithm effectively.

REFERENCES


Process Model Consistency Measurement

Sukanth Sistla & A.Ananda Rao
Department of CSE, JNTUA College of Engineering, Ananthapur, India
E-mail : sukanth.886@gmail.com, akepogu@gmail.com

Abstract – In software development, managers increasingly focus on process improvement which in turn has increased the demand for software metrics. To turn the business application of a software project to system specification is a big challenge in business environment. Since business analysts and system analysts have their own perspective, the modeling of business process is necessary to facilitate both the perspectives and for better coordination. An attempt is made to resolve inconsistencies and to develop model that verifies process consistency described in Petri net graph. From this model it has been confirmed that the consistency ranges from 0.0 to 1.0. In addition to the above Support Vector Machines (SVM) usage is helpful to improve consistency with greater confidence to evaluate behavioral and structural consistency.

Keywords - change propagation, consistency checking, behavioral profiles, behavioral equivalence, process model.

I. INTRODUCTION

To eliminate the gap[2][9] between business applications and system specifications, business analysts and system analysts have their own perspective needed to be coordinated properly, many applications of such business processing model[1] have given raise to problems and maintaining consistency of such related models has become a challenge for business modeling practice. Behavioral profile is a solution to the inappropriateness of behavioral notions and also change propagation between models including inconsistencies can be resolved. Through this model free-choice Petri nets[4] with reference to their places and transitions, profiles can be computed. Schema integration[5] in particular schema matching investigates and shows such correspondences can be identified automatically. Methodologies for integrated system design like matching techniques and graphical matching can also be applied. Targeting research challenge of defining a notion of consistency between process models[3] is more adequate than existing notions of behavioral equivalence. Behavioral profiles are less sensitive to projections than trace equivalence of as behavioral profiles remain unchanged even if start and end branches are introduced. Profile consistency[1][3] ranging from 0 to 1.0. The proposed change uses Support vector machines can to improve consistency with greater confidence.

II. CONSISTENCY MEASUREMENT USING BEHAVIORAL PROFILES AND STRUCTURAL ANALYSIS

Business process change[10] is at the very core of business process management, reaching from business evolution to process enactment, results in multiple models that overlap in content due to serving different purposes. That, in turn, imposes serious challenges for the propagation of changes between these process models.

2.1 Process Models

Our notion of a process model[3][8] is based on a graph containing activity nodes and control nodes, which, in turn, captures the commonalities of process description languages.

Thus, the subset of BPMN used in our initial example can be traced back to the following definition of a process model.

Definition 1: (Process Model) A process model[10] is a tuple P (A, I,C,F,T) with A as a non-empty set of activity nodes, I as a set of initial activities, F as the flow relation, and T : {and,or,xor} as a function that assigns a type to control nodes.
2.2 Behavioral Profile

The Behavioral profile[1] aims at capturing Behavioral aspects of a process in a fine-grained manner. That is, it consists of three relations between nodes of a process graph. These relations are based on the notion of weak order.[3]. Thus, weak order does not have to hold for all traces of the model.

- The strict order relation \( x \succ P(y) \land y \prec \succ P(x) \)
- The exclusiveness relation \( x \prec \succ P(y) \land y \prec \succ P(x) \)
- The observation concurrency relation \( x \succ P(y) \lor y \succ P(x) \)

The set of all three[8] relations is the Behavioral profile.

Two process models with equivalent behavioral profiles may differ in the trace equivalence, in contrast the two process models with identical trace equivalence can also identical in behavioral profiles.

Correspondence Relation: if the relation between two process models is left unique and is not functional

Projected Firing sequence: In a sequence considered, the set of aligned sequences is referred as firing sequence.

Trace Consistency of Alignment: If Aligned transitions of a projected firing sequence contain trace equivalence then it reflects as Trace consistency of alignment.

2.3 Structural Analysis

The structural analysis of dynamic lumped process models[8] forms an important step in the model building procedure and it is used for the determination of the solvability properties of the model, too. This analysis includes the determination of the degree of freedom, structural solvability, differential index and the dynamic degrees of freedom. As a result of the analysis, the decomposition of the model is obtained and the calculation path can be determined. This way the appropriate numerical method for solving the model can be chosen efficiently. Moreover, advice on how to improve the computational properties of the model by modifying its form or its specification can also be given.

Effective graph-theoretical methods have been proposed in the literature based on the analysis tools developed by, for the determination of the most important solvability property of lumped dynamic models: the differential index. The properties of the dynamic representation graph of process models described by semi-explicit DAE-systems have also been analyzed there in case of index 1 and higher index models. Beside the algorithm of determining the differential index by using the representation graph, a model modification method has also been proposed in the literature, which results in a structurally solvable model even in the case of higher index models.

2.4 Dynamic Representation Graph

A dynamic graph is a sequence of static graphs corresponding to each time step of the integration. On a dynamic graph there are directed arcs attached from the previous static graph to the succeeding static graph that are determined by the method applied for solving the ordinary differential equations. In case of a single step explicit method, the value of a differential variable at time \( t+\) is computed using the corresponding differential value and its value at a previous time \( t \). For example, when the explicit Euler method is used:

\[
x(t + h) = x(t) + h.x'(t) \rightarrow eq(1)
\]

where \( h \) denotes the step length during the numerical integration. The structure of a dynamic graph assuming explicit Euler method for solving differential equations is shown in figure 1.

![Dynamic Graph model for Euler method for solving differential](image)

Fig. 1: Dynamic Graph model for Euler method for solving differential

The structural analysis based on graph theoretical technique is carried out in steps performed sequentially. The first step is to rewrite the model into its standard form. The second step is the assignment of types to vertices in the representation graph. The important types of vertices determined by the model specification are the following:

- **<S>(set)-type variables**: These represent variables, which are assigned to the specified given values. In the case of a dynamic representation graph assuming explicit method for solving the differential equations, the differential variables will be labelled by type `<S>` because
their initial value can be obtained from the initial values, and then their values can be calculated step by step by numerical integration. Labels \(<S>\) and \(<S^*>\) are treated the same way during the analysis.

- \(<G>\text{(given)}\)-type variables: A variable assigned to a specific value of a left hand side is a \(<G>\)-type variable. Unlike the \(<S>\)-type variables, the values of the right hand side variables will be suitably adjusted so as to preserve the equality of the two sides.

III. CONSISTENCY MEASURES FOR ALIGNED PROCESS MODELS

The previously defined concept of a behavioral profile[1][3] allows us to formally discuss the notion of a degree of profile consistency between a pair of process models.

3.1 Consistency based on Behavioral Profiles

![Exemplary alignment setting](image)

In general, our notion of consistency based on behavioral profiles, i.e., profile consistency, is grounded on the preservation of behavioral relations for corresponding activities. In contrast to the notion of a trace consistent alignment, it does not require the correspondence relation to be injective. Instead, it allows for 1:n (and even n:m) correspondences. Therefore, this notion can be applied to vertical as well as horizontal alignments. Preservation of the behavioral relation is only required in case there are no overlapping correspondences. With respect to the examples in Figure 2, it is easy to see that all pairs of aligned transitions are also consistent with respect to their behavioral relation. For instance, the strict order relation between transitions A and D in model 2(a) is preserved for transition pair A and D1, as well as A and D2 in model 2(c). In addition, in all three models it holds C \(\parallel\) C. That is, C might occur multiple times during execution.

3.2 Interpretation of Profile Consistency

As exemplified in the previous section, the degree of profile consistency ranges between 0 and 1.0 for two process models and a correspondence relation. Still, a degree of 1.0 does not imply that both models are (projected) trace equivalent. This stems from the fact that the underlying behavioral profile represents a behavioral abstraction; apparently, the degree of profile consistency quantifies the quality of an alignment with respect to the order of potential activity occurrences. A degree of 1.0 guarantees that all these constraints are equal for the aligned activities of two models. A degree of 0.9, in turn, indicates that the constraints on the order of potential activity occurrences are equal solely for 90% of the relations between aligned activities. As the degree of profile consistency measures the quality of the alignment, its definition is independent of the coverage of the process models by the correspondence relation (i.e., the share of activities in both models that are aligned). Based on the degree of profile consistency, consistency thresholds might be defined. However, we assume these thresholds to be highly dependent on a specific project setting. Once a degree of profile consistency below 1.0 is observed, the question of how to locate the source of inconsistency has to be addressed. According to our approach, inconsistencies manifest themselves in different relations of the behavioral profile of two process models for a pair of aligned activities. This information can directly be provided to business analysts and system analysts in order to judge on the necessity of the inconsistency. While this kind of feedback allows for locating the inconsistency directly in case of only a few inconsistent profile relations (e.g., caused by an interchanged order of two activities in a sequence), it might be inappropriate if a big number of profile relations is inconsistent.

IV. EXPERIMENTS AND RESULTS ANALYSIS

After preprocessing of the benchmark models, we are able to analyze its consistency. As mentioned before, we establish correspondences between events and functions with equal labels. Further on, we extract all pairs of process models that are aligned by at least two correspondences. For such a pair, we then calculate the consistency measures, that is, trace consistency, the degree of trace consistency, and the degree of profile consistency of the alignment and finally analyzed the accuracy of the degree of profile using structural analysis.
The results are optimistic from the experiments conducted on benchmark business models represented in PetriNet format. We consider the consistency measurement systems WF systems (WF) [6], and Behavior profiling (BP) analysis to compare with the proposed Behavior Profiling and Structural Analysis (BP&SA) [8]. We can find the significant benefit of BP&SA over other models [7]. Figure 2 represents the comparison of optimality in consistency measurement between BP&SA and other two models [7]. In figure 3 we can observe the computational overhead of the WF [6]. Here BP is having slight advantage over BP&SA, which can be negligible while considering the accuracy achieved through BP&SA in consistency measurement.

TABLE 1: Overall Results

<table>
<thead>
<tr>
<th>Technique</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical N-M without stemming</td>
<td>0.72</td>
<td>0.60</td>
<td>0.68</td>
</tr>
<tr>
<td>Lexical N-M with stemming</td>
<td>0.72</td>
<td>0.60</td>
<td>0.66</td>
</tr>
<tr>
<td>A-Star with Post Processing</td>
<td>0.81</td>
<td>0.60</td>
<td>0.69</td>
</tr>
<tr>
<td>Greedy</td>
<td>0.89</td>
<td>0.60</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Fig. 2: Optimality in Consistency Measurement

V. CONCLUSIONS

Process models play an important role to reduce the gap between business requirements and system specifications. In this article, we have discussed addressed the research challenge of defining a notion of consistency between process models that is more adequate to this problem than existing notions of behavioral equivalence. Behavioral profiles are used for the definition of the formal notion of profile consistency. Behavioral profiles provide three major advantages in contrast to the existing notion of trace equivalence and consistency measures that build up it. First, behavioral profiles are less sensitive as behavioral profiles remain unchanged even if additional start and end branches are added. Second, the structure of a behavioral profile provides degree of profile consistency ranging from 0 to 1.0 and Structural analysis accurate the consistency measurement through degree of profile. Finally, the concept of a behavioral profile builds informal properties of free-choice Petri nets. We proved that profile consistency can be checked for sound free-choice WF-systems in O(n3) time with n nodes. and to evaluate behavioral and structural consistency measure SVM is helpful.

REFERENCES


Automation of Media Access Cross Connect

Giridhar & Siddesh G M
Department of Information Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, India
E-mail : giritrivedi@gmail.com, siddeshgm@gmail.com

Abstract – Sanity testing is the process of testing the basic functionality of any software product. Sanity testing would take considerable amount of time by test engineer to test the product. If sanity testing itself fails, then the effort put on other testing techniques would go waste, as the build is declared un-useable if Sanity fails. Hence it is a good idea to automate sanity testing process so that, same sanity suite can be run as soon as a new build arrives. This automation tool is developed for a specific product, Media Access Cross Connect. The tool is being divided into two parts. First part, records all test cases in the background, whenever a test engineer performs sanity testing and saves it in XML file format. Second part, replays same configurations using a script and carries out verification and validation for the response data to decide success or failure of the test cases. It is record once run many times activity.

Keywords - XML, PERL, WebCT, JSON, url, HTML, DOM, GUI.

I. INTRODUCTION

Test automation tools have changed the way, testing is performed in this era. Testing is an important part of software development life cycle. If problems come up during implementation phase, delivery time of the product slides. This will add more pressure on testers, leading to inefficient testing or delaying the product release or more resources need to be allocated to testing team to meet aggressive deadlines [7]. Automation tools will reduce this time required for testing. Advantage of this automation tool is two fold. First, test engineers need not have to re-create failed test case scenarios as they would have already present in XML file. Second, developers can refer to saved XML file to find out, locality of occurrence of problems without contacting test engineers. One can rely to certain extent, on automation tools to perform testing even if the delivery time of the product slides. Our automation tool intends to help perform sanity testing. This tool belongs to the category of record and replay automation tools. This tool is being divided into two parts; first part captures all configurations in background, done by test engineer. Second part replays configurations captured in XML file. Verification and validation is carried out for response data. If result shows as successful then, execution of the test case is considered to be passed, else it is considered to be failed. Verification and validation is done to ensure that, software products can satisfy all requirements and certain performance [2].
format. This JSON response is decoded by PERL packages, then verification and validation for all parameters present in the XML file is done.

II. RELATED WORK

Ben-menachem M and Marless G.S in [6] presents, a minor defective component can cause major adverse effects if the developed software is not thoroughly tested prior to its real-world implementation. Hence the tester has to test all components within the limited time under the pressure of delivery dates. Automating testing mechanism helps reduce time required for testing and improve the quality of product. According to [4] there are several types of test automation, namely, Test management, Unit test, Test data generation, performance test, functional/system/ regression test. Popular test automation category is “Test Execution” automation technology also known as capture/playback or record/playback. [5] calls software testing as the indispensable phase in modern software life cycle. Software testing is done to ensure that the developed software is useful and can run properly in real world applications. Automation is considered to be, the future trend of testing. The current practice of software test automation is based on recording manual test activities and replaying recorded test scripts for sanity and regression testing. It is imperative to reduce cost and improve effectiveness of software testing by automation testing process which contains many testing related activities using various techniques and methods [3]. Software automation can improve the productivity and quality of work to a great extent [2].

```xml
<root>
  <serverType>
    <REQ>
      <POST> URL </POST>
      <Param1> name = value </Param1>
      <Param2> name = value </Param2>
      <Param3> name = value </Param3>
      <Param4> name = value </Param4>
    </REQ>
    <TMDLY> time </TMDLY>
    <STEPID> id </STEPID>
  </REQ>
</root>
```

Table 1: XML file Format

III. RECORDING TOOL

Recording tool is implemented in java script. This tool will be used as an add-on to either Mozilla Firefox or Google chrome browser. As on today recording tool works fine with all the versions of Firefox browsers and it works only on one version of Google chrome browser. This tool takes help of an open source tool’s source code just to capture POST methods generated, whenever test engineer tries to configure the server either through webCT or through a browser.

![Fig 2: Recording Tool](image_url)

Block diagram of recording tool is shown in fig 2. Product for which this tool is developed, contains different names on the GUI than that of corresponding names present in JSON response present on sever. Once all configurations have been done, test engineer clicks on save button, a POST method will be generated. This POST method will be saved and underlying DOM structure of underlying page is recursively iterated upwards to find the “form” tag of page. This form tag will give us underlying HTML form name. This HTML form will be opened and searched, to find out JSON parameter names present on GUI. This form tag contains a return block inside it. Return block contains mapping for fields present on form and corresponding JSON response. We will extract whole block at a time, search only for parameters present in POST method for verification and validation. All these validation and verification parameters along with POST method will be put in “REQ” block of XML file. In addition to all these, a TMDLY tag representing time delay will also be inserted into the “REQ” block. The tag STEPTYPE indicates, type of testing is being performed. Likewise for every configuration done, a separate “REQ” block will be generated. This XML file acts as a test case. Collections of such XML files form a Test Suite.

The pseudo code used to iterate through the DOM structure of the file is as shown in Algorithm 1.

```
WHILE (parentNode != Null)
  Obtain element’s parent node.
  IF parent’s node = FORM
    Extract innerHTML of parent 2 level above
    Extract “return” block
  END IF
END WHILE
```

International Conference on Computer Science and Information Technology, ISBN: 978-93-81693-86-5, 10th June, 2012-Tirupati
FOR(each line in return "Block")
    Separate JSON and form name
ENDFOR
return JSON and form name
ENDWHILE

Algorithm 1: Parse JSON parameters

For read-only parameters present on the GUI, the same procedure will be repeated to extract the underlying JSON parameters except the upward iteration would be done till the "div" tag is found.

IV. PLAYBACK TOOL

Playback tool is implemented in PERL. PERL script takes captured XML file as a reference to replay all configurations present in "REQ" block. XML file will have to be parsed so that, we will have well defined data structures present in the PERL. This parsing helps us to play around the converted data for verification and validation at a later point in time.

The high level working of this playback tool is as follows. PERL script takes XML file as input. Reads each “REQ” block present in XML file. This “REQ” block contains all information pertaining to configurations done and various parameters to be verified and validated. These configurations will be replayed. Server will return configured data or response, in JSON format. This JSON data is again decoded into PERL data structures using available JSON decoders in PERL. The block diagram of the playback tool is shown in fig 3.

![Fig 3: Replay Tool](image)

We make use of XML parsers to parse the XML files, which contain test cases. There are basically two types of XML parsers, SAX parser and BNF parser [1]. PERL packages are used to parse this XML file into PERL hashes. The package used is XML::Simple. This package will help to parse XML file, returns a reference to converted PERL hashes. Data::Dumper package, available in PERL is used to dump converted hashes into some file or on to standard output. Returned reference from parser will be pointing to one among, hash, array of hashes or to scalars. We need to find out to which data structure the reference is pointing to. That can be done using following algorithm 2

FOR (eachkey in HASH )
    obtain the reference pointing to that key
    IF ref = HASH
        It is a hash
    ELSE IF ref = ARRAY
        It is an array
    ELSE
        It is a scalar
ENDIF
ENDFOR

Algorithm 2: To Parse JSON Response.

Once XML file is parsed, each block in the XML file will be considered as a single test step. This will be replayed using LWP::UserAgent package of PERL. There are two methods available in this package, GET and POST. This will return reference to response. When this reference is made to point to "contents" field, actual response will be available. Simplest use of this method is as shown in table 2.

$UserAgent = LWP::UserAgent->new();
$response = $UserAgent->get($URL);
$sresponse = $UserAgent->post($URL);

Table 2: UserAgent Package Usage

Suseragent is new instance of LWP::UserAgent package. This has to be created first, before performing above actions. The response obtained like this will be in the form of JSON. This response has to be decoded for further verification and validation of parameters present in the “REQ” block of XML file.

Response in JSON format will be decoded using JSON::PP package. Procedure to decode is as in table 3.
$str = $json->allow_barkey->decode($fileContents)

Table 3: Decode JSON Response

$json is the new instance created for that package. $str contains reference to the response. Decode method will decode the response into definite PERL data structures. Reference would be pointing to scalar, hash, array, array of hashes, hash of hashes or any other combination available. This response is compared against the key-value pair of verification and validation parameters present in each REQ block of XML file. If all key-value pairs found to be present in response, then that test step is considered to be passed. If any one key-value pair is not present, then test step is considered to be failed. If any of the test steps present in XML file is said to be failed, then whole test case is considered to be a failed test case.

V. PERFORMANCE EVALUATION

The configuration on which the tool is being used or tested contains a computer system with minimum of 512MB RAM, 40 GB Hard Disk, Media Access Cross Connect, WebCT or Browser, PERL Interpreter.

“Time” is one parameter which can be used for evaluating the performance of the automation tool. Here we compare the time and effort taken by the test engineer to execute certain number of test steps against the time taken by the automation tool. It is found that as the number of test steps increases, the time taken for manual testing also keeps increasing while the time taken by automation tool is far less than of the manual time. It is shown in graph 1.

“Success Rate” is another parameter used to check stability of tool. 0 to 4 test steps in every 100 test steps are failing as on today. Once this tool is developed completely, we are expecting all the test steps to pass through. The graph of “success rate” is as shown in graph 2.

VI. CONCLUSION

Sanity testing process can be well automated to save time of a test engineer, which can be used for other useful purpose. This will set us the benchmark to automate other testing techniques too. Automation tool under development works fine with the product for which it is developed. Currently the time required for testing by one man week hour is being reduced to only two hours. The technology used in this tool can also be used for other products with little modification, which suits for the GUI structure designed for the product.

The tool is in the testing phase, once testing is completed the enhanced version of the tool is released with improved “success rate”.

REFERENCES:


Node Mobility Pattern Model for Location Updates

K. Jagadeeswari & K. Madhavi
CSE Department, JNTUA College of Engineering, Anantapur, Andhra Pradesh, India
E-mail: jaga.konduru@gmail.com, kasamadhavi@yahoo.com

Abstract – We contemplate the location service during a mobile ad-hoc network (MANET), where every node has to maintain its location data by 1) frequently updating its location data at intervals its neighboring region, that is named neighborhood update (NU), and 2) often updating its location data to bound distributed location server within the network, that is named location server update (LSU). The tradeoff between the operation prices in location updates and therefore the performance losses of the target application owing to location inaccuracies (i.e., application costs) imposes a vital question for nodes to make a decision the optimal strategy to update their location data, where the optimality is within the sense of minimizing the general prices. During this paper, we have a tendency to develop a model that pattern the location information based on the moving directions of the nodes. In the location detection phase, the proposed model finds movement patterns based on local trajectories. To address the energy conservation issue in resource-constrained environments, the algorithm only transmits the local grouping results to the sink node for further validation. In the group validation phase, our model combines the local grouping results to derive the group relationships from a global view. We further leverage the mining results to track the locations of the moving nodes efficiently. The results of experiments show that the proposed model achieves good grouping quality, and the pattern technique helps reduce the energy consumption by reducing the amount of data to be transmitted.

I. INTRODUCTION

With the advance of terribly large-scale integrated circuits (VLSI) and also the industrial popularity of worldwide positioning services (GPS), the geographic location data of mobile devices during a mobile ad-hoc network (MANET) is becoming accessible for varied applications. This location data not solely provides an added degree of freedom in planning network protocols [1], however is also crucial for the success of the many military and civilian applications [2], [3], e.g., localization in future battlefield networks [4], [5] and public safety communications [6], [7]. In a MANET, since the locations of nodes aren't fastened, a node must frequently update its location data to some or all different nodes. There are 2 basic location update operations at a node to keep up its up-to-date location data within the network [8]. One operation is to update its location data inside a neighboring region, where the neighboring region isn't essentially restricted to 1 hop neighboring nodes [9], [10]. we tend to decision this operation neighborhood update (NU), that is sometimes implemented by native broadcasting/flooding of location data messages.

The other operation is to update the node’s location data at one or multiple distributed location servers. The positions of the situation servers can be fastened (e.g., Home zone-based location services [11], [12]) or unfixed (e.g., Grid Location Service [13]). we tend to decision this operation location server update (LSU), that is sometimes implemented by unicast or multicast of the situation data message via multihop routing in MANETs. It's obvious that there's a tradeoff between the operation prices of location updates and also the performance losses of the target application within the presence of the situation errors (i.e., application costs). On one hand, if the operations of NU and LSU are too frequent, the facility and communication bandwidth of nodes are wasted for those unnecessary updates. On the opposite hand, if the frequency of the operations of NU and/or LSU isn't sufficient, the situation error can degrade the performance of the appliance that depends on the situation data of nodes (see [3] for a discussion of various location accuracy strategies needs for various applications). Therefore, to reduce the general prices, location update ways got to be rigorously designed. Typically speaking, from the network purpose of read, the optimal style to reduce overall prices ought to be jointly dispensed on all nodes, and thus, the ways can be coupled.

However, Such a style features a formidable implementation complexity since it needs data concerning all nodes, that is difficult and dear to get. Therefore, a additional viable style is from the individual node purpose of read, i.e., every node independently chooses its location update strategy with its native data. during this paper, we offer a stochastic
call framework to investigate the situation update downside in MANETs, we tend to formulate the situation update downside at a node as a Markov call method (MDP) [8], beneath a widely used Markovian mobility model [9], [10], rather than solving the MDP model directly, the target is to spot some general and demanding properties of the matter structure and also the optimal answer that might be useful in providing insights into sensible protocol style. we tend to 1st investigate the answer structure of the model by identifying the monotonicity properties of optimal NU and LSU operations with respect to (w.r.t.) location inaccuracies beneath a general value setting. Then, given a separable value structure such that the results of location inaccuracies induced by insufficient NU operations and LSU operations are separable, we tend to show that the situation update choices on NU and LSU is independently dispensed while not loss of optimality, i.e., a separation property exists.

From the discovered separation property of the model and also the monotonicity properties of optimal actions, we discover that 1) there forever exists an easy optimal threshold-based update rule for LSU operations where the brink is mostly location dependent; 2) for NU operations, an optimal threshold-based update rule exists during a heavy-traffic and/or a low-mobility situation. The separation property of the matter structure and also the existence of optimal thresholds in LSU and NU operations, not solely considerably simplify the search of optimal location update ways, however conjointly offer tips undesigning location update algorithms in observe, we tend to conjointly offer a sensible model-free learning approach to seek out a near-optimal answer for the situation update downside, within the case that no a priori data of the MDP model accessible in observe. Up to our data, the situation update downside in MANETs has not been formally addressed as a stochastic call downside. The theoretical work on this downside is additionally terribly restricted. In [9], the authors analyze the optimal location update strategies, in terms of minimizing achievable overall routing overhead. Although, closed-form optimal update thresholds obtained in [9], it's solely valid for his or her routing theme.

II. VARIABLE LENGTH MARKOV CHAIN MODEL (VLMC) AND CASUISTIC SUFFIX HIERARCHY (CSH)

If the movement of a node is regular then the next location of that node is possible to predict based on its preceding locations. We leverage the temporal-and-spatial correlations of the moving node and use a VLMC to model the statistics. Under this model, a node's movement is modeled by the conditional probability distribution over \( \Sigma \) for a given location sequence data set. Specifically, for a location sequence \( s \) over \( \Sigma \) and a symbol \( \sigma \in \Sigma \), \( P(\sigma | s) \) is the conditional probability that \( \sigma \) will follow \( s \). The length of \( s \) is floating, which provides the flexibility to adapt to the variable length of movement patterns.

Among many implementations of VLMC, CSH [4] has the lowest storage requirement [15]. The CSH building algorithm extracts significant patterns from a data set, prunes unnecessary nodes during Hierarchy construction, and then generates a CSH. Each node in the Hierarchy is labeled by a string \( s \), which represents a significant pattern with occurrence probability above the minimal threshold \( P_{\text{min}} \). Each node carries the conditional empirical probabilities, i.e., \( P(\sigma | s) \), for every \( \sigma \in \Sigma \), and the maximal length of \( s \) is specified by \( L_{\text{max}} \).

The CSH algorithm has an excellent capacity for extracting structural information from sequences. Its low complexity, i.e., \( O(\ell) \) in both time and space [7], makes it more attractive to be used in streaming or resource-constrained environments. Compared with algorithms that mine all accurate frequent patterns, the compact Hierarchy structure and the controllable size of a CSH are particularly useful in resource-constrained environments. For example, if the conditional probabilities of a pattern "ABCD" are similar to those of "BCD," CSH will only store "BCD." Moreover, CSH is efficient in predicting the occurrence probability of a sequence or predicting the next symbol of a sequence. The occurrence probability of a sequence \( s \) regarding to a CSH \( T \), denoted by \( P^T(s) \), is the prediction of the occurrence probability of \( s \) based on \( T \). For example, given a CSH \( T \), as shown in Fig. 2, the occurrence probability \( P^T(\text{"nokjfb"}) \) is computed as follows:

\[
P^T(\text{"nokjfb"}) = P^T(\text{"n"}) \times P^T(\text{"o"}) \times P^T(\text{"k"}) \times P^T(\text{"n"}) \times P^T(\text{"o"})
\]

\[
\times P^T(\text{"j"}) \times P^T(\text{"f"}) \times P^T(\text{"b"}) \times P^T(\text{"n"})
\]

\[
P^T(\text{"n"}) \times P^T(\text{"o"}) \times P^T(\text{"k"}) \times P^T(\text{"o"})
\]

\[
\times P^T(\text{"j"}) \times P^T(\text{"f"}) \times P^T(\text{"b"}) \times P^T(\text{"n"})
\]

\[
= 0.05 \times 1 \times 1 \times 1 \times 1 \times 0.33
\]

\[
= 0.0165.
\]

For a given sequence \( s \) and a CSH \( T \), our predict next algorithm outputs the next most likely symbol \( a \).
We explain the algorithm by an example to demonstrate the efficiency of the algorithm. Given a sequence \( s = \text{nokjf} \)

![Fig. 1 The framework of our distributed mining algorithm:](image)

At the local end, CH performs the NMPM Mine algorithm to generate local grouping result \( G_i \) while the sink performs the Cluster Ensembling algorithm to combine the local grouping results into a consensus final result \( G_\delta \) and a CSH.

**III. MINING NODE MOBILITY PATTERNS**

In this work, we model the movement of a node by a VLMC, and use a CSH to mine the significant movement patterns. The advantages of CSH include its computing and storage efficiency as well as the information it carries. In the tracking application, nodes are tracked periodically so that the time interval of consecutive items of a location sequence is implied. The CSH building algorithm scans the sequence for significant movement patterns, whose items are constrained to be consecutive in the location sequence. This is also why the computing cost is much lower than sequence pattern mining. Moreover, a CSH provides us important information in similarity comparison. For a pattern and a CSH, we can predict the occurrence probability of the pattern, which is viewed as the importance of the pattern regarding the CSH.

A set of moving nodes is regarded as belonging to the same group if they share similar movement patterns. In this section, we first propose a new similarity measure to define the pair wise similarity of moving node. The advantages of the new proposed similarity measure \( \text{simp} \) include its efficiency and its accuracy. First, \( \text{simp} \) compares the

\[ T, \text{ as shown in Fig. 2, the predict}_\text{next} \text{ algorithm traverses} T \text{ to the deepest node} \text{node}_\text{okjf} \text{ based on} s. \text{ The path includes} \text{node}_\text{root}, \text{node}_\text{def}, \text{node}_\text{eff}, \text{node}_\text{diff}, \text{and} \text{node}_\text{def}. \text{ Finally, symbol} \text{ 'e', which has the highest conditional empirical probability in} \text{node}_\text{diff}, \text{ is returned as the most probable next symbol. Since the algorithm's computational overhead is limited by the height of a CSH, it is suitable for sensor nodes. Further details about the CSH building algorithm and related discussions about parameter setting can be found in [9], [2]. The notations and key symbols used in the paper are}

---

<table>
<thead>
<tr>
<th>Notation/Symbol</th>
<th>Description</th>
<th>Notation/Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D = {o_0, o_1, \ldots, o_N} )</td>
<td>A set of objects</td>
<td>( D )</td>
<td>A set of thresholds (CE)</td>
</tr>
<tr>
<td>( G = {g_0, g_1, \ldots, g_m} )</td>
<td>A grouping result containing ( m ) groups of objects</td>
<td>( \delta )</td>
<td>A threshold in the range ([0,1])</td>
</tr>
<tr>
<td>( C = {G_0, G_1, \ldots, G_K} )</td>
<td>An ensemble of ( K ) local grouping results</td>
<td>( G_\delta )</td>
<td>The identified group ID of object ( o_n )</td>
</tr>
<tr>
<td>( S )</td>
<td>A finite alphabet representing IDs of a cluster of sensors</td>
<td>( \delta )</td>
<td>Jaccard Similarity Coefficient</td>
</tr>
<tr>
<td>( P(s) )</td>
<td>A location sequence of length ( l ), where ( \forall o \in S )</td>
<td>( TC )</td>
<td>The transmission cost</td>
</tr>
<tr>
<td>( \text{Patoc} )</td>
<td>The empirical conditional probability that ( o ) is right after ( s )</td>
<td>( N )</td>
<td>The number of objects of interest</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>A PST, ( \gamma ) is the PST associated with ( s )</td>
<td>( K )</td>
<td>The number of random-walk objects</td>
</tr>
<tr>
<td>( \text{Layer} )</td>
<td>The maximal memory length of a PST</td>
<td>( r )</td>
<td>The radius of an SG in hops</td>
</tr>
<tr>
<td>( \text{Pre} )</td>
<td>The predicted occurrence probability of ( o ) regarding ( s )</td>
<td>( \text{layer} )</td>
<td>Layer number of the network structure</td>
</tr>
<tr>
<td>( \text{simmax} )</td>
<td>The new proposed similarity measure</td>
<td>( \ast )</td>
<td>Square root of the size of a sensor cluster</td>
</tr>
<tr>
<td>( \text{simmin} )</td>
<td>The minimal similarity threshold (GMPMine)</td>
<td>( EB )</td>
<td>A specified error bound</td>
</tr>
<tr>
<td>( S )</td>
<td>The union of nodes in two PSTs</td>
<td>( \text{GDR} )</td>
<td>The group dispersion radius</td>
</tr>
</tbody>
</table>
listed in Table 1. Similarity of two nodes based on their significant movement patterns instead of their entire location sequences. In [14], a variation of CSH, named emission Hierarchy, is used to train the patterns in a streaming environment. simp can be directly applied to the mature nodes of two emission Hierarchies, instead of all nodes. Thus, simp can provide efficiency for the applications with evolving and evolutionary similarity relationships.3 Second, it considers the importance of each movement pattern regarding to each individual node so that it achieves better accuracy in similarity comparison.

With the definition of simp, two nodes are similar if their similarity score is above a minimal threshold. A set of nodes is regarded as a group if each node is similar to at least half the members of the same group. To tackle the problem of discovering groups of moving nodes, we propose a distributed mining algorithm comprised of a NMPM algorithm and a CV&R algorithm as shown in Fig. 3. The NMPM algorithm uses a CSH to generate the significant movement patterns and computes the pair wise similarity of moving nodes by using simp. It utilizes the HCS algorithm to cluster the moving nodes into non-overlapped groups. To address the energy consumption problem in WSNs, our algorithm only transmits the local grouping result into non-overlapped groups. To address the energy consumption problem in WSNs, our algorithm only transmits the local grouping result.

3.1 Similarity Measurement

In this work, we use a CSH to mine significant movement patterns of a node, where a significant movement pattern is a subsequence with occurrence probability higher than a minimal threshold. Each node of a CSH represents a significant movement pattern and carries its conditional probabilities, and all nodes of a CSH provide the precise information about the predicted occurrence probability of a given pattern.

To provide better discrimination accuracy, we propose a new similarity measure simp that adequately and skillfully utilizes the information carried by CSHs to measure the similarity of two nodes. The design concepts of simp are simple and useful. Different from the previous works, such as [15], that equal weights the patterns, we further consider importance and difference of a pattern related to each individual node. The importance of a pattern s is modeled by using the predicted occurrence probability, i.e., $P^T(s)$, while the difference of a pattern is defined over all of the dimensions, i.e., $P^T(\sigma|s)$, $\forall \sigma \in \Sigma$. Based on the two concepts, we define the distance a pattern s associated with two nodes $o_i$ and $o_j$ as

$$d(s) = \sqrt{\sum_{\sigma \in \Sigma} (P^T(o_i) - P^T(o_j))^2}$$

$$= \sqrt{\sum_{\sigma \in \Sigma} (P^T(o_i) \times P^T(\sigma|s) - P^T(o_j) \times P^T(\sigma|s))^2},$$

where $T_i$ and $T_j$ are their respective CSHs. $d(s)$ is the euclidian distance of products of the importance and difference over $\Sigma$ related to $o_i$ and $o_j$. Note that since the similarity of two nodes is symmetric in our applications, a symmetric measure, such as euclidean distance, is more desirable. Furthermore, for a pattern $s \in T_i$, $P^T_i(s)$ is a significant value because the occurrence probability of $s$ is higher than the minimal support $Pmin$. If $o_i$ and $o_j$ share the pattern $s$, we have $s \in T_i$ and $s \in T_j$ respectively, such that $P^T_i(s)$ and $P^T_j(s)$ are non-negligible and meaningful in the similarity comparison. Thus, we define the similarity score of $o_i$ and $o_j$ by using all of their significant movement patterns as follows:

$$simp(o_i, o_j) = -\log \frac{\sum_{\sigma \in \Sigma} (P^T_i(\sigma|s) - P^T_j(\sigma|s))^2}{2L_{max} + \sqrt{2}},$$
3.2 The Node Mobility Pattern Model (NMPM) Algorithm

We now describe the Node Mobility Pattern Model algorithm, which identifies groups of nodes and determines their movement patterns. Fig. 3 illustrates the NMPM algorithm, where $S$ represents the location sequence data set and $N$ denotes the number of nodes of interest. The minimal similarity threshold ($\text{sim}_{\min}$) is the lower limit of the similarity between two nodes belonging to the same group. Let $O = \{o_0, o_1, \ldots, o_N\}$ denote the nodes of interest and $\pi(o_i)$ denote the mapping of the group ID and node $o_i$. The NMPM algorithm generates the grouping result $G$ and the associated group movement patterns $GT$. Specifically, $G$ is composed of $m$ disjoint groups of nodes over $O$, denoted by $G = \{g_0, g_1, \ldots, g_{m-1}\}$, where $g_i = \{o_j | \pi(o_j) = i, o_j \in O\}$. The group movement patterns associated $g_i$ is denoted by $GT_i$, and $GT = \{GT_0, GT_1, \ldots, GT_{m-1}\}$ denotes the group movement patterns for the $m$ groups.

The NMPM algorithm is comprised of four steps. First, we extract the movement patterns of each node from the location sequence. Second, we construct a similarity graph in which similar nodes are connected by an edge. Third, we extract highly connected components to derive the group information. Fourth, we construct a group CSH for each group in order to conserve the memory space.

**Simulations and Results Discussion**

We implemented a simulation model of a ad hoc network with node mobility using MXML and action-script to evaluate the performance of our design. In the experiments, we use the VLM mobility model to synthetically generate location data, i.e., the coordinates $(x, y)$, for a group of nodes. The location-dependent mobility model [13] is used to simulate the roaming behavior of a group leader. The other member nodes are followers that are uniformly distributed within a specified group dispersion radius ($GDR$) of the leader, where the $GDR$ is the maximal hop count between the followers and their leader. We utilize the $GDR$ to control the dispersion degree of the nodes. A smaller $GDR$ implies stronger group relationships, i.e., nodes are
closer together. In addition, we control the movement range of a group of nodes by using the movement distance \((d)\), which is the linear distance between the starting point and the furthest point reached by the leader node. A longer distance implies that nodes move across more sensor clusters. To ensure our simulation reflects the real-world scenarios, we input nodes with a random-walk model. The simulation results revealed the benefit of the NMPM over stochastic location updates (SLU) model [15]. We can find the advantage of NMPM over SLU in terms of resource utilization in fig 4(a). The fig 4(b) indicates the NMPM scalability in computational overhead by retaining the location update optimality, which is similar to SLU that can find in fig 4(c).

**IV. CONCLUSIONS**

In this work, we exploit the characteristics of group movements to discover the information about groups of moving nodes in mobile ad hoc network. In contrast to the stochastic location update technique, we mine the group information in a distributed manner. We propose a novel model, which consists of a local NMPM algorithm and a CV&R algorithm, to discover group information. Our algorithm finds node movement patterns as well as group information and the estimated group dispersion radius. Using the found node movement patterns and the group information, we design an energy-efficient location update strategy. The contribution of our approach is threefold: 1) it reduces energy consumption by allowing LUs to avoid sending the prediction-hit locations, because the locations can be recovered by the sink via the same prediction model; 2) it leverages group information in data aggregation to eliminate redundant update traffic; and 3) it sets the size of an SG adaptively to limit the amount of flooding traffic. Our experimental results show that the proposed mining technique achieves energy saving in location update process, which retained the optimality in location information with fewer computational overhead.

**REFERENCES**


..........
INTRODUCTION

Image Processing is a technique to enhance raw images received from cameras/sensors placed on satellites, space probes and aircrafts or pictures taken in normal day-to-day life for various applications. Various techniques have been developed in Image Processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned spacecrafts, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel computers, large size memory devices, graphics softwares etc. Image Processing is used in various applications such as:

- Remote Sensing
- Medical Imaging
- Non-destructive Evaluation
- Forensic Studies
- Textiles
- Material Science.
- Military
- Film industry
- Document processing
- Graphic arts
- Printing Industry

The common steps in image processing are image scanning, storing, enhancing and interpretation. The schematic diagram of image scanner-digitizer diagram is shown in fig 1.

Methods of Image Processing

There are two methods available in Image Processing.

Analog Image Processing:

Analog Image Processing refers to the alteration of image through electrical means. The most common example is the television image.

The television signal is a voltage level which varies in amplitude to represent brightness through the image. By electrically varying the signal, the displayed image appearance is altered. The brightness and contrast controls on a TV set serve to adjust the amplitude and reference of the video signal, resulting in the brightening, darkening and alteration of the brightness range of the displayed image.

Digital Image Processing:

In this case, digital computers are used to process the image. The image will be converted to digital form using a scanner – digitizer [6] (as shown in Figure 1).
and then process it. It is defined as the subjecting numerical representations of objects to a series of operations in order to obtain a desired result. It starts with one image and produces a modified version of the same. It is therefore a process that takes an image into another.

The term digital image processing generally refers to processing of a two-dimensional picture by a digital computer [7,11]. In a broader context, it implies digital processing of any two-dimensional data. A digital image is an array of real numbers represented by a finite number of bits.

The principle advantage of Digital Image Processing methods is its versatility, repeatability and the preservation of original data precision.

The various Image Processing techniques are:

- Image representation
- Image preprocessing
- Image enhancement
- Image restoration
- Image analysis
- Image reconstruction
- Image data compression

**Image Representation:**

An image defined in the "real world" is considered to be a function of two real variables, for example, \( f(x,y) \) with \( f \) as the amplitude (e.g. brightness) of the image at the real coordinate position \( (x,y) \). The effect of digitization is shown in Figure 2.

![Image 2](image2.png)

**Fig. 2**

The 2D continuous image \( f(x,y) \) is divided into \( N \) rows and \( M \) columns. The intersection of a row and a column is called as pixel. The value assigned to the integer coordinates \( [m,n] \) with \( \{m=0,1,\ldots,M-1\} \) and \( \{n=0,1,\ldots,N-1\} \) is \( f[m,n] \). In fact, in most cases \( f(x,y) \) which we might consider to be the physical signal that impinges on the face of a sensor. Typically an image file such as BMP, JPEG, TIFF etc., has some header and picture information. A header usually includes details like format identifier (typically first information), resolution, number of bits/pixel, compression type, etc.

**Image Preprocessing**

**Scaling**

The theme of the technique of magnification is to have a closer view by magnifying or zooming the interested part in the imagery. By reduction, we can bring the unmanageable size of data to a manageable limit. For resampling an image Nearest Neighborhood, Linear, or cubic convolution techniques [5] are used.

i. **Magnification**

This is usually done to improve the scale of display for visual interpretation or sometimes to match the scale of one image to another. To magnify an image by a factor of 2, each pixel of the original image is replaced by a block of 2x2 pixels, all with the same brightness value as the original pixel.

![Image 3](image3.png)

**Fig. 3**: Image Magnification

ii. **Reduction**

To reduce a digital image to the original data, every \( m \)th row and \( m \)th column of the original imagery is selected and displayed. Another way of accomplishing the same is by taking the average in 'm x m' block and displaying this average after proper rounding of the resultant value.

![Image 4](image4.png)

**Fig. 4**: Image Reduction
**Rotation**

Rotation is used in image mosaic, image registration etc. One of the techniques of rotation is 3-pass shear rotation, where rotation matrix can be decomposed into three separable matrices.

3-pass shear rotation

\[
R = \begin{vmatrix}
\cos \alpha & -\sin \alpha \\
\sin \alpha & \cos \alpha \\
1 - \tan \alpha / 2 & 0 & 1 - \tan \alpha / 2 \\
0 & 1 & \sin \alpha & 1 & 0 & 1
\end{vmatrix}
\]

**Advantages**

1. No scaling – no associated resampling degradations.
2. Shear can be implemented very efficiently.

**Mosaic:**

Mosaic is a process of combining two or more images to form a single large image without radiometric imbalance. Mosaic is required to get the synoptic view of the entire area, otherwise capture as small images.

**Image Enhancement Techniques:**

Some times images obtained from satellites and conventional and digital cameras lack in contrast and brightness because of the limitations of imaging sub systems and illumination conditions while capturing image. Images may have different types of noise. In image enhancement, the goal is to accentuate certain image features for subsequent analysis or for image display[1,2]. Examples include contrast and edge enhancement, pseudo-coloring, noise filtering, sharpening, and magnifying. Image enhancement is useful in feature extraction, image analysis and an image display. The enhancement process itself does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application-dependent.

Some of the enhancement techniques are

1. Contrast Stretching
2. Noise Filtering
3. Histogram modification

1. **Contrast Stretching**

Some images(eg water bodies , deserts, dense forests, snow, clouds and under hazy conditions over heterogeneous regions) are homogeneous i.e., they do not have much change in their levels. In terms of histogram representation, they are characterized as the occurrence of very narrow peaks. The homogeneity can also be due to the incorrect illumination of the scene.

Ultimately the images hence obtained are not easily interpretable due to poor human perceptibility. This is because there exists only a narrow range of gray-levels in the image having provision for wider range of gray-levels. The contrast stretching methods are designed exclusively for frequently encountered situations. Different stretching techniques have been developed to stretch the narrow range to the whole of the available dynamic range.
2. Noise Filtering:

Noise filtering is used to filter the unnecessary information from an image. It is also used to remove various types of noises from the images. Mostly this feature is interactive. Various filters like low pass, high pass, mean, median etc., are available.

![Fig. 8 : Noise Removal](image1)

3. Histogram Modification:

Histogram has a lot of importance in image enhancement. It reflects the characteristics of image. By modifying the histogram, image characteristics can be modified. One such example is Histogram Equalization. Histogram equalization is a nonlinear stretch that redistributes pixel values so that there is approximately the same number of pixels with each value within a range. The result approximates a flat histogram. Therefore, contrast is increased at the peaks and lessened at the tails.

![Fig. 9 : Edge Enhancement](image2)

Image Analysis:

Image analysis is concerned with making quantitative measurements from an image to produce a description of it [8]. In the simplest form, this task could be reading a label on a grocery item, sorting different parts on an assembly line, or measuring the size and orientation of blood cells in a medical image. More advanced image analysis systems measure quantitative information and use it to make a sophisticated decision, such as controlling the arm of a robot to move an object after identifying it or navigating an aircraft with the aid of images acquired along its trajectory. Image analysis techniques require extraction of certain features that aid in the identification of the object. Segmentation techniques are used to isolate the desired object from the scene so that measurements can be made on it subsequently. Quantitative measurements of object features allow classification and description of the image.

Image Segmentation:

Image segmentation is the process that subdivides an image into its constituent parts or objects. The level to which this subdivision is carried out depends on the problem being solved, i.e., the segmentation should stop when the objects of interest in an application have been isolated e.g., in autonomous air-to-ground target acquisition, suppose our interest lies in identifying vehicles on a road, the first step is to segment the road from the image and then to segment the contents of the road down to potential vehicles. Image thresholding techniques are used for image segmentation.

Classification:

Classification is the labeling of a pixel or a group of pixels based on its grey value [9,10]. Classification is one of the most often used methods of information extraction. In Classification, usually multiple features are used for a set of pixels i.e., many images of a particular object are needed. In Remote Sensing area, this procedure assumes that the imagery of a specific geographic area is collected in multiple regions of the electromagnetic spectrum and that the images are in good registration. Most of the information extraction techniques rely on analysis of the spectral reflectance properties of such imagery and employ special algorithms designed to perform various types of ‘spectral analysis’. The process of multispectral classification can be performed using either of the two methods: Supervised or Unsupervised.

In Supervised classification, the identity and location of some of the land cover types such as urban, wetland, forest etc., are known as priori through a combination of field works and toposheets. The analyst attempts to locate specific sites in the remotely sensed...
data that represents homogeneous examples of these land cover types. These areas are commonly referred as TRAINING SITES because the spectral characteristics of these known areas are used to 'train' the classification algorithm for eventual land cover mapping of reminder of the image. Multivariate statistical parameters are calculated for each training site. Every pixel both within and outside these training sites is then evaluated and assigned to a class of which it has the highest likelihood of being a member.

In an Unsupervised classification, the identities of land cover types has to be specified as classes within a scene are not generally known as priori because ground truth is lacking or surface features within the scene are not well defined. The computer is required to group pixel data into different spectral classes according to some statistically determined criteria.

The comparison in medical area is the labeling of cells based on their shape, size, color and texture, which act as features. This method is also useful for MRI images.

**Image Reconstruction from Projections**

Image reconstruction from projections [3] is a special class of image restoration problems where a two-(or higher) dimensional object is reconstructed from several one-dimensional projections. Each projection is obtained by projecting a parallel X-ray (or other penetrating radiation) beam through the object. Planar projections are thus obtained by viewing the object from many different angles. Reconstruction algorithms derive an image of a thin axial slice of the object, giving an inside view otherwise unobtainable without performing extensive surgery. Such techniques are important in medical imaging (CT scanners), astronomy, radar imaging, geological exploration, and nondestructive testing of assemblies.

**Image Compression**

Compression is a very essential tool for archiving image data, image data transfer on the network etc. They are various techniques available for lossy and lossless compressions. One of most popular compression techniques, JPEG (Joint Photographic Experts Group) uses Discrete Cosine Transformation (DCT) based compression technique. Currently wavelet based compression techniques are used for higher compression ratios with minimal loss of data.
REFERENCES


