<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Identifying Prime Testing Zones To Ensure Quality ETL Routine Of A Data Warehouse</td>
<td>1586</td>
</tr>
<tr>
<td>Jaiteg Singh , Kawaljeet Singh</td>
<td></td>
</tr>
<tr>
<td>2. Comparative Performance Analysis of UI Test Automation from Various Databases for New Web Applications</td>
<td>1601</td>
</tr>
<tr>
<td>G.Appasami , K.Suresh Joseph</td>
<td></td>
</tr>
<tr>
<td>3. Improved Reliability In MANET’s Using Enhanced Double Coverage Broad Casting (EDCB)</td>
<td>1611</td>
</tr>
<tr>
<td>4. An Efficient Dynamic Router Approach To Defeat DDOS Attack S.S.Nagamuthu Krishnan, V.Saravanan</td>
<td>1622</td>
</tr>
<tr>
<td>5. Blocking Parameter Driven Virtual Topology Reconfiguration for IP-Over-WDM Networks with QOS Parameters Ramasamy Mariappan, R.Shanmugalakshmi</td>
<td>1630</td>
</tr>
<tr>
<td>6. Performance Computation Of Object Oriented Programs Through UML Dr. Vipin Saxena and Deepa Raj</td>
<td>1638</td>
</tr>
<tr>
<td>7. Software Maintainability Prediction Model for Object-Oriented Software Systems Based On Sensitivity-Based Linear Learning Method Sunday Olusanya Olatunji</td>
<td>1645</td>
</tr>
<tr>
<td>9. FPGA Based Quadrature Decoder/Counter with FC Bus Interface For Motor Control Manu. T. S , B.Thilagavathi and Anithamary.X</td>
<td>1677</td>
</tr>
</tbody>
</table>
Identifying Prime Testing Zones to Ensure Quality ETL Routine of a Data Warehouse

Jaiteg Singh1  Kawaljeet Singh2

1Research Scholar, University College of Engineering, Punjabi University, Patiala - 147002, India. Email : jaitegkhaira@yahoo.co.in
2Director, University Computer Centre, Punjabi University, Patiala - 147002, India. Email : singhkawaljeet@pbi.ac.in

ABSTRACT
Attaining high quality accurate data for decision making is a major challenge for the data warehouse industry. The organizations practicing data warehouse are diverse in their functionality and follow varying business practices. To tackle the data quality and data management issues which are inherent in data warehouse implementation, the project team has to develop an extensive reviewing process for improving data quality. A thorough business practice review, an analysis of existing data quality and data cleansing techniques are necessary to develop a data quality assurance methodology. The quality assurance for an in house developed the extraction transformation and loading (ETL) routine using hand coded algorithms is not an easy task. This low level implementation of ETL logic may cause hidden errors at technical as well as at logical level. Hence instigation of automated testing procedures as an integrated module of ETL routine itself is essential to ensure a high quality data in a data warehouse. The identification of such crucial test cases which may be automated to enhance data quality in a data warehouse is the sole aim of this paper.

Keywords : ETL, Data Warehouse, ELT, ETL Prototype

INTRODUCTION
The importance of high quality data with respect to the strategic planning of any organization cannot be ignored. The Data Warehousing Institute, (TDWI), in a recent report, estimates that data quality problems currently cost U.S. businesses about $600 billion each year. Generally the benefits of high quality data are sidelined because of the expenses associated with attaining high quality data. A data quality strategy is very important specially when implementing a data warehouse. Although the effectiveness of a data warehouse is based upon the quality of its data but the data warehouse does not do a satisfactory job of cleansing data. The same data would need to be cleansed repeatedly during iterative operations. The best place to cleanse data is the ETL platform. By cleansing data in ETL instead of in the data warehouse, organizations can save time and money.

Methodology Followed : Precise data is an elementary condition for effective data warehouse development. Most data warehouses and information systems contain momentous quantity of imprecise data. There is plenty of literature available on the ETL structure, data quality and data warehousing but there is very diminutive information available on the quality assurance of ETL routines. Hence the authors followed an empirical approach to identify prime test cases essential for a quality ETL development and to understand the ETL routine structure itself. Initially an ETL prototype was developed (snapshot 1 to snapshot 5) which is capable of extracting data from a number of distinct databases following different structures and formats. After the analysis of
extracted data a number of test cases were developed to assure the quality data in the targeted data warehouse.

The organizations lack fundamental awareness of the concepts of information quality. The shortage of accurate data costs organizations dearly in correction actions along with lost customers, missed opportunities, and incorrect decisions. Most organizations are very much ignorant of the enormity of the data cleansing costs because they are unaware of the extent of inaccurate data in their database systems. Organizations are content in their belief that their data is good enough, although they have no basis for that belief [1]. In case of a data warehouse the prime responsibility for the data quality is of the ETL routine which is responsible to extract, transform and load data from different sources into the data warehouse. Hence the selection of an appropriate ETL tool is a serious concern of an organization. There are a number of ETL tools available in the market but generally small and medium sized enterprises use a hand coded ETL routine to extract and unify data and the expenditure associated with a licensed commercial ETL tool can also be avoided.

The ETL is a big term having many small independent sub systems of its own like:

1. **Aggregate Building System:** It is responsible for generating and maintaining physical database structures, known as aggregates. These are used to improve query performance. It includes stand-alone aggregate tables and materialized views.

2. **Backup System:** This system is responsible for back up, metadata, recovery, restart, security, and compliance requirements.

3. **Cleansing System:** This system is usually a dictionary driven system and is generally responsible for parsing generic details of individuals and organizations etc. It can identify and remove the duplicate records and retains back references like natural keys to all participating original sources.

4. **Data Change Identification System:** It keeps an eye over Source log file readers, source date and sequence number filters etc.

5. **Data Match Up:** Its sole responsibility is data integration across multiple data sources based upon their conformed attributes and measures.

6. **Data Profiling System:** This system is responsible for the analysis of Column properties of the source tables including detection of dependent domains, and structure analysis. It further handles foreign keys and candidate keys also.

7. **Error Handler:** It is a widespread system for identifying and retorting to all ETL error events. It has special routines to handle various classes of errors, and includes real-time monitoring of ETL data quality.

8. **Error Tracking System:** It is an automatic as well as a manual system for trapping and resolution of an error condition. It includes simple error log entries, operator notification, and messages for the system developer.

9. **Fact Table Loader:** It is a System for updating transaction fact tables including manipulation of indexes and partitions. Normally used to append most recent data.

10. **Job Schedule Handler:** This System is for scheduling and launching all ETL jobs. It is able to halt itself for a wide variety of system conditions including dependencies of prior jobs completing successfully. It can also post alerts.

11. **Jumble Dimension Handler:** It is responsible for the creation and maintenance of dimensions
consisting of miscellaneous low cardinality features and indicators found in most production data sources.

12. Late Arriving Dimension Handler: This system is responsible for the insertion and update of dimension changes that have been deferred in arriving at the data warehouse.

13. Late Arriving Fact Handler: This system has an insertion and update logic for fact records that have been delayed in arriving at the data warehouse.

14. Meta Data Assembler: It assembles the metadata context associated with each fact table and loads it to the fact table as a normal dimension.

15. Metadata Repository Manager: This is an indispensable system of an ETL routine for capturing and maintaining all ETL metadata and transformation logic.

16. Multi Valued Dimension Associative Table Builder: This module is responsible for creating an associative table used for describing many-to-many relationship between dimensions.

17. Multidimensional Cube Builder: It is responsible for creation and maintenance of multidimensional (OLAP) cubes, including special preparation of dimension hierarchies as dictated by the specific cube technology.

18. Pipelining System: This system is highly desirable to automatically invoke pipelining system for any ETL process to tackle certain conditions, such as not writing to the disk in case of transaction failure or waiting on a condition in the middle of the query under execution.

19. Quality Checker: This system is responsible to check the quality of all the incoming flows of data.

20. Recovery and Restart system: This is a common system which is responsible for restarting a job that has halted.

21. Report and Dependency Analyzer: It can watch the ultimate physical sources and all subsequent transformations of any selected data element, chosen either from the middle of the ETL pipeline, or chosen on a final delivered report. It can record all affected downstream data elements and final report fields which are affected by a potential change in any selected data element.

22. Security System: This system is responsible for the security of data within an ETL pipeline and the ETL itself.

23. Slowly Changing Dimension Handler: This system has transformation logic for handling of time variance for a dimension attribute which generally includes events like overwrite, create new field or create new record.

24. Source Extract System: This system includes Source data adapters along with push/pull routines for filtering and sorting at the source. Such system is also responsible for data format conversions, and data staging after transfer to ETL environment.

25. Surrogate Key Pipeline: It is a Pipelined, multithreaded process for replacing natural keys of incoming data with data warehouse surrogate keys.

26. Version Control System: This system is required for archiving and recovering all the metadata in the ETL pipeline. It keeps a vigil on all the check-outs and check-ins of all ETL modules and jobs. This system has source comparison capability to reveal differences between different versions.

27. Workflow Monitor: This system includes a control panel and reporting system to watch all job runs...
Identifying Prime Testing Zones to Ensure Quality ETL Routine of a Data Warehouse

initiated by the Job Scheduler. It records number of records processed, summaries of errors, and actions taken etc.

Extract Transform and Load (ETL) was believed to be the most effectual way to insert data into a data warehouse. Early data warehouses ETL systems were not proficient of managing the extensive processing required to perform the complex transformations involved in the warehouse load process. So third-party tools like IBM’s WebSphere DataStage and Informatica were used to organize data movement between source systems and the data warehouse [8]. Now a days with the advancement in both hardware and data warehouse development technology, the designers now consider Extract Load and Transform (ELT) as a better approach instead of extract transform and load (ETL). Because of data explosion the organizations have to manage pentabytes of data which may require hours or even days for ETL process completion depending upon the amount of data to be extracted and the complexity of the transformation rules.

In the case of ETL, before passing the data along to the data warehouse the data is moved to an intermediate platform where the transformation rules are applied. Where as the ELT follows a standard data transfer mechanism such as File Transfer Protocol (FTP) to transfer the bulk data directly to the data warehouse. The transformation rules are then applied and the data warehouse tables instead of any intermediate staging area. The difference between both these architectures is shown in figure 1 and figure 2 respectively.

Figure 1 : The ETL Architecture

Figure 2 : The ELT Architecture

Both these architectures are responsible for fetching and merging data from different sources into the data warehouse. The only factor that differentiates the duo is the staging area. In ETL architecture the extracted data is first placed onto a staging area where the transformation logics are applied to consolidate as well as filter data to enhance its quality before loading this purified data into the data warehouse [13]. In case of ELT there is no staging area the data is loaded directly into the data warehouse. The transformations are applied within the data warehouse server itself. However after the rigorous analysis of both these architectures the following characteristics have been identified:

Table 1 : ETL Vs ELT

<table>
<thead>
<tr>
<th></th>
<th>ETL</th>
<th>ELT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A dedicated external system is applied to take care of transformation logic for data standardization and business rules thus reducing the unnecessary burden from the data warehouse.</td>
<td>There is no dedicated external system responsible to tackle transformation logic and business rules. The transformations are done on the data already loaded into the data warehouse.</td>
</tr>
<tr>
<td>2</td>
<td>The whole data has to travel first from source to staging area and then from staging area to the data warehouse through the network thus causing excess network</td>
<td>The files are loaded from the source systems to the data warehouse via FTP or other secure file transfer methods, hence the network traffic is</td>
</tr>
</tbody>
</table>
traffic when there is no dedicated link between the ETL server and the data warehouse.

3 The ETL server requires high performance CPU and huge disk capacity to sustain the transformation process. This can lead to the need for expensive and highly sophisticated hardware.

ETL tools have the capability to interact with other external engines for data validation before the data is loaded on the data warehouse, such as Geographic Information Systems (GIS).

Errors if any that occur during the transformation process can be located and corrected before data is loaded in the data warehouse table thus reducing the need for time consuming database roll-backs.

Complex transformations which may require external sources of data are not easy to implement with the stored procedures of the data warehouse.

Database roll-backs are inevitable in case an error occurs during the transformation process. Generally these rollbacks are taken on temporary tables.

The cost for loading the data warehouse is quite lower than the ETL architecture as there is no additional software licence is required.

Time for getting data to the data warehouse is reduced as there is no staging process required.

Before selecting the loading procedure one must weigh up the data transformation requirements along with the desired data quality of the targeted database. If the transformation rules are intricate and cannot be carried out using stored procedures of the database than ELT architecture should be avoided and in case where routines require text parsing ELT architecture are excellent for data standardization and cleansing [12]. For hefty environments where more than ten source systems are feeding the data warehouse and one terabyte or greater of transactional data is involved, the ETL is the best suited architecture. On the other hand ELT is best suited for loading small data sets where relatively simple transformation logic is applied. ELT is also best suited for manipulating business data for populating data marts that has a physical infrastructure similar to that of the data warehouse.

As business organizations investigate their data unification needs, the first decision they need to make is whether to build or buy an ETL tool. Although the ETL tools offered by various vendors are very much proficient in their functionality but still there are many organizations that believe it is better to hand write ETL programs than use off-the-shelf ETL software. These companies advocate their decision by aiming at the high cost of many ETL tools and the profusion of programmers on their staff. Although it is very crucial decision but to facilitate this build or buy decision the following aspects can be considered.

<table>
<thead>
<tr>
<th>Build</th>
<th>Buy</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is cheaper and quicker to code ETL programs than use a vendor ETL tool.</td>
<td>ETL tools are expensive to purchase and requires renewal of license.</td>
</tr>
</tbody>
</table>

<p>| Table 2 : Build Vs Buy Analysis |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>It is cheaper to maintain as it is geared with a specific business.</td>
</tr>
<tr>
<td>3</td>
<td>Code written is based upon custom specifications and meta data model.</td>
</tr>
<tr>
<td>4</td>
<td>No need to pay unnecessary training or maintenance fee to any vendor.</td>
</tr>
<tr>
<td>5</td>
<td>Easily available object oriented technology is best suited for ETL development.</td>
</tr>
<tr>
<td>6</td>
<td>To keep the costs down and for better turnaround times one can outsource the ETL development code to Asian countries.</td>
</tr>
<tr>
<td>7</td>
<td>Challenging factors like migrating source data into a data warehouse along with data cleansing jobs can easily be performed with the hand coded ETL routine.</td>
</tr>
<tr>
<td>8</td>
<td>Meta data is rarely maintained.</td>
</tr>
<tr>
<td>9</td>
<td>These tools are flexible and can be adjusted in accordance with the changing business dimensions.</td>
</tr>
<tr>
<td>10</td>
<td>Complex mappings can be handled easily with custom built ETL code.</td>
</tr>
<tr>
<td>11</td>
<td>It is difficult to ensure adequate stability, reliability and performance.</td>
</tr>
<tr>
<td>12</td>
<td>Team of expert programmers is required to develop a customized ETL tool.</td>
</tr>
<tr>
<td>13</td>
<td>Creating and integrating user defined functions is a cumbersome and difficult job.</td>
</tr>
<tr>
<td>14</td>
<td>Rigorous testing and debugging effort is required.</td>
</tr>
<tr>
<td>15</td>
<td>Source data is well understood in advance to tackle a particular database.</td>
</tr>
</tbody>
</table>

These tools are not much flexible and one has to look in for ready made plug ins to cope with changing business dimensions. Only predefined mapping procedures can be carried out which generally elude complex mappings. Adequate stability, reliability and performance is well ensured in advance. Highly salaried and experienced programmers are already employed by ETL vendors for developing, training and maintenance purposes. The well designed proficient modules are integrated in advance and one has to study the user manual only to make it work. No need of testing and debugging only maintenance is required. They follow a generalized approach in understanding source databases.
The use of different software systems from different vendors in order to provide full coverage of the business functions and the integration between these systems often requires real-time interaction among them [5]. The easiest way to data migration is with the help of in-house built solutions with hand-coded algorithms for Extraction, Transformation and Load (ETL). This low-level implementation makes the maintenance of such migration solutions a complicated job and can be a cause for hidden errors with the ETL processes on logical or technical level and once it has been decided to build an ETL project there begins a new chore of assuring its quality. Accurate data does not come free. It requires careful attention to the design of ETL systems, constant monitoring of data collection activities and assertive actions to correct problems that generate or propagate inaccurate data. Any hand coded ETL module can not be used on real-life data management until and unless its performance has been assured by testing it rigorously [6,10]. A successful test case is one which makes the system halt at the occurrence of a particular event. To identify prime testing zones for a hand-coded ETL tool the authors coded an ETL tool which is capable of extracting data from databases like Oracle, SQL, MS Access, MS Excel and flat files like those of MS Word. This tool is capable of making predefined transformations and data purification to the extent possible for managing the personal records database. However, one can write test cases for the click of a button but it will merely be a test case which will not prove anything fruitful. Hence in this paper only those test cases have been discussed in Table 3, which are essential to enhance the quality of an ETL routine of a data warehouse.
Table 3: Prime Test Cases for ETL Quality Assurance

<table>
<thead>
<tr>
<th>S.No</th>
<th>Test Case Desc.</th>
<th>Input</th>
<th>Expected Outcome</th>
<th>Actual Result</th>
<th>Assigned to</th>
<th>Defect Severity *</th>
<th>Result (P/F)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Connecting to the Source Database</td>
<td>ETL routine tries to connect the targeted Source database</td>
<td>Source Database is connected</td>
<td>There is no connectivity of the targeted source database</td>
<td>Development team</td>
<td>Major</td>
<td>Fail</td>
<td>Provide the source and destination address carefully if in a networked environment provide the source and destination IP address along with the port no to be opened, also check if a firewall is interrupting the connection</td>
</tr>
<tr>
<td>2</td>
<td>Availability of the Source Database</td>
<td>ETL routine tries to extract source data</td>
<td>Source data is available every time the ETL tries to extract data</td>
<td>Source data is available at particular times</td>
<td>Development team</td>
<td>Major</td>
<td>Fail</td>
<td>Parley for a time slot with the source DBA and break the extraction into smaller chunks</td>
</tr>
<tr>
<td>3</td>
<td>Availability of the Source Database</td>
<td>ETL routine tries to extract source data</td>
<td>Source database is swiftly available through ODBC OLEDB connections</td>
<td>Target source database requires a particular driver because we are unable to use ODBC OLEDB connections</td>
<td>Development team</td>
<td>Major</td>
<td>Fail</td>
<td>Acquire or get hold of database drivers and install on ETL server</td>
</tr>
<tr>
<td>4</td>
<td>Availability of the Source Database</td>
<td>ETL routine tries to extract source data</td>
<td>Source data server name is accepted</td>
<td>The target source server name is not accepted</td>
<td>Development team</td>
<td>Major</td>
<td>Fail</td>
<td>Check for the specified name of the source database, some databases like teradata requires a suffix to the name and if on a networked environment try it with the help of IP address else ask the network administrator to add the name on system DNS</td>
</tr>
<tr>
<td>5</td>
<td>Availability of the Source Database</td>
<td>ETL routine tries to extract source data</td>
<td>Source data is available and the ETL is capable of querying the database</td>
<td>Targeted source database is available but ETL is unable to query the database</td>
<td>Development team</td>
<td>Major</td>
<td>Fail</td>
<td>The source DBA should manage two accounts, an admin account for the development purpose and a functional account for ETL routines.</td>
</tr>
<tr>
<td></td>
<td>Availability of the Source Database</td>
<td>Source OLTP database is called in</td>
<td>Source OLTP database version is in accordance with the drivers installed on the ETL server.</td>
<td>The database version of the OLTP is newer than the driver on the ETL server</td>
<td>Development team</td>
<td>Minor</td>
<td>Fail</td>
<td>Upgrade the driver.</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------</td>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------------</td>
<td>------</td>
<td>-----</td>
<td>---------------------</td>
</tr>
<tr>
<td>7</td>
<td>Data Extraction from a flat file</td>
<td>File is imported</td>
<td>Structure of the file is justified and bulk data is imported from the file</td>
<td>Structure of file is not justified</td>
<td>source DBA</td>
<td>Minor</td>
<td>Fail</td>
<td>specify the source system administrator about the desired structure of the file system.</td>
</tr>
<tr>
<td>8</td>
<td>Data Extraction from a flat file</td>
<td>File is imported and file is cleared from archive directory</td>
<td>Bulk data is imported from the file while cleaning the source archive directory.</td>
<td>Data has been extracted but replica is still existing on archive directory.</td>
<td>Development team</td>
<td>Minor</td>
<td>Fail</td>
<td>Decide an archive period keeping in view the backup procedure, reprogram the ETL routine for the shifting of extracted files to archive directory and to delete archived files older than the archive period.</td>
</tr>
<tr>
<td>9</td>
<td>Data Extraction from a flat file</td>
<td>Data is to be extracted from the file</td>
<td>Order and number of columns extracted are same as anticipated.</td>
<td>Number of columns are more than expected, the order of columns is also disturbed</td>
<td>Development team</td>
<td>Minor</td>
<td>Fail</td>
<td>Avoid the usage of semicolons, colons, commas and tabs etc. file delimiters. Use some uncommon characters as delimiters like ~ (tilde) and pipes.</td>
</tr>
<tr>
<td>10</td>
<td>Data Extraction from a flat file</td>
<td>Data is to be extracted from the file</td>
<td>Order and number of columns extracted are same as anticipated.</td>
<td>Number of columns are more than expected, the order of columns is also disturbed</td>
<td>Development team</td>
<td>Minor</td>
<td>Fail</td>
<td>Prefer column to column mapping. Try to preserve the order of columns and agree with the source DBA on the number of columns which are going to be extracted.</td>
</tr>
<tr>
<td>11</td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
<td>Source database schema is in accordance with warehouse schema.</td>
<td>There is a mismatch in the count of attributes defined for a single entity in the source schema and in the target schema.</td>
<td>Development team and source DBA</td>
<td>Minor</td>
<td>Fail</td>
<td>Arrange a meeting with the source DBA and decide a mutual agreed upon schema.</td>
</tr>
<tr>
<td>Table No.</td>
<td>Description</td>
<td>Details</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Extracting Relational Databases</td>
<td>There is no well-defined criteria to define the primary key. The primary key is hard to identify.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Extracting Relational Databases</td>
<td>Data is extracted iteratively.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Extracting Relational Databases</td>
<td>The source columns are modified. The ETL routine maintains the source database.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Extracting Relational Databases</td>
<td>The ETL routine updates only the source database.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Extracting Relational Databases</td>
<td>The ETL routine updates only the source database.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>Source DML</td>
<td>Violation of the primary key.</td>
</tr>
<tr>
<td>19</td>
<td>Source DML</td>
<td>There is no well-defined criteria to define the primary key. The primary key is hard to identify.</td>
</tr>
<tr>
<td>20</td>
<td>Source DML</td>
<td>The source columns are not modified. The ETL routine maintains the source database.</td>
</tr>
<tr>
<td>21</td>
<td>Source DML</td>
<td>The ETL routine updates only the source database.</td>
</tr>
<tr>
<td>22</td>
<td>Source DML</td>
<td>The ETL routine updates only the source database.</td>
</tr>
<tr>
<td>23</td>
<td>Source DML</td>
<td>The ETL routine updates only the source database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>The Development team</td>
<td>ETL approaches the relational database</td>
</tr>
<tr>
<td>25</td>
<td>The Development team</td>
<td>There is no well-defined criteria to define the primary key. The primary key is hard to identify.</td>
</tr>
<tr>
<td>26</td>
<td>The Development team</td>
<td>The source columns are not modified. The ETL routine maintains the source database.</td>
</tr>
<tr>
<td>27</td>
<td>The Development team</td>
<td>The ETL routine updates only the source database.</td>
</tr>
<tr>
<td>28</td>
<td>The Development team</td>
<td>The ETL routine updates only the source database.</td>
</tr>
<tr>
<td>29</td>
<td>The Development team</td>
<td>The ETL routine updates only the source database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>31</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>32</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>33</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>34</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>35</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>37</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>38</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>39</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>40</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>41</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>43</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>44</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>45</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>46</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td>47</td>
<td>Count of source values</td>
<td>The count of source values is equal to the count of values in the target database.</td>
</tr>
<tr>
<td></td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>18</td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
</tr>
<tr>
<td>19</td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
</tr>
<tr>
<td>20</td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
</tr>
<tr>
<td>21</td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
</tr>
<tr>
<td>22</td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
</tr>
<tr>
<td></td>
<td>Extracting Relational Databases</td>
<td>ETL approaches the relational database</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>23</td>
<td>Verifying ETL Functionality</td>
<td>User tries to access ETL staging area</td>
</tr>
<tr>
<td>24</td>
<td>Verifying ETL Functionality</td>
<td>The transformation logics are applied to the staging area</td>
</tr>
<tr>
<td>25</td>
<td>Verifying ETL Functionality</td>
<td>ETL tries to update Target Database</td>
</tr>
<tr>
<td>26</td>
<td>Verifying ETL Functionality</td>
<td>ETL tries to back up data in case of update failure</td>
</tr>
<tr>
<td>27</td>
<td>Verifying ETL Functionality</td>
<td>The backed up data has to be recovered</td>
</tr>
</tbody>
</table>
In the above said prototype the authors have automated all these prime test cases along with many others. The results produced by this prototype were satisfactory as the quality of resulting database improved considerably. It has further been observed that if quality checks are imposed during the data extraction stage than the effort required to refine and transform data can be reduced considerably resulting in the saving of time and money. Business houses are investing heavily on new generation databases so as to gain competitive edge. As with any development project, a data warehouse also needs testing. The complexity and size of data warehouse systems make
Identifying Prime Testing Zones to Ensure Quality ETL Routine of a Data Warehouse

comprehensive testing both “more difficult and more necessary”. The fact, queries that perform satisfactorily on small datasets may fail miserably in the real life environment. This necessitates establishing a system that runs queries on fully scaled data. The scarcity of this fully scaled test data is again a crucial problem hence one may use a test data generator tool for generating synthetic test data.

CONCLUSION & FUTURE WORK
The primary reason for the convolution of the data extraction and transformation functions are the diversity of the source systems. This diversity includes bewildering combination of computing platforms, operating systems, database management systems, network protocols, and source legacy systems etc. Hence there is a need to pay special attention to the various sources and begin with generating a complete record of the source systems. With this record as a starting point one should work out all the details of data extraction. The difficulties encountered in the data transformation function should also be related to the heterogeneity of the source systems. The loading procedure might seem to be the simplest one but it is solely responsible for consolidation and integration of targeted database. Although the authors have observed the enhancement in the data quality of the target database but still the statistical analysis of the results is under process and the authors are hopeful to statistically prove the impact of ETL on data quality of a data warehouse.

REFERENCES


Author’s Biography

Jaiteg Singh is a research scholar from Punjabi University Patiala, Punjab, India. His area of research includes Data Warehousing and Testing.

Dr. Kawaljeet Singh is Director, UCC, Punjabi University Patiala, Punjab, India. His area of research includes system simulation and Data Warehousing.
Comparative Performance Analysis of UI Test Automation from Various Databases for New Web Applications

G. Appasami1     K. Suresh Joseph2

ABSTRACT

Silverlight applications are used to create platform independent, browser independent interactive and attractive web applications with Dot net supporting languages. User Interface (UI) Test Automation is mainly used to reduce the time, cost and manual work. There are several methods available like keyboard driven, mouse driven for UI Test Automation. In this paper we made performance analysis of various set of UI Test Automation for Silverlight applications. UI Test Automation is more difficult for web based applications when compared with windows based applications. UI accessibility and automation are very difficult for Silverlight applications. In this paper we analyzed different types of UI Test automation with different kinds of input data and output data. In this paper we analyzed the performance of various input data for various UI test automation.

Keywords : User Interface, Test Automation, Automation Peer and Silverlight Applications, Performance Analysis.

1. INTRODUCTION

Silverlight is Microsoft’s new cross browser or delivering richer interactive applications to users over the web. Silverlight 2.0 is Microsoft’s second release of the Silverlight platform [18][19]. Silverlight 2’s biggest change from Silverlight 1.0 is the inclusion of a compact version of the .NET Framework, complete with the .NET Framework 3.0 Common Language Runtime. By adding .NET to Silverlight, Microsoft makes it easy for .NET developers to reuse their existing programming skills, collaborate with designers, and quickly create rich applications for the Web. One of the key benefits of Silverlight 2 is that it can execute any .NET language, including C# and VB.NET [17][18]. Silverlight 2 ships with a “lightweight” version of the full .NET Framework, which features, among other classes, extensible controls, XML Web Services, networking components, and LINQ APIs. This class library is a subset of the .NET Framework’s Base Class Library, which enables the Silverlight plug-in to be a fast and small download. In addition to the .NET Framework classes, Silverlight 2 also ships with a subset of the WPF UI programming model, including support for shapes, documents, media, and WPF animation objects [15][18][19].

Testing these kinds of new applications with existing old testing methods are ridicules [1][3][4]. We have to test new applications developed by new technology and also reduce test time and cost.

Microsoft is developed Silverlight and provides accessibility classes [15][21]. Automation Peer UI Test approach is best to do test automation for Silverlight applications [19][22].

1Dept. Of Comp. Science, Pondicherry Univ. Email : appas_9g@yahoo.com

2Lect. Dept. of CS, Pondicherry Univ. Email : sureshjosephk@yahoo.co.in
2. USER INTERFACE MARKUP LANGUAGES FOR A WEB APPLICATION

Ajax, XUL, XAML, Flex, platforms are emerging, each one with its advantages and its drawbacks, and the choice is posed since one wants to build a Web application, a RIA (Rich Internet Application) having the same interface with attractive and interactive user interfaces. We listed all widely used (for graphical user interface) markup languages, and we have to choose the best one for our needs.

A. GladeXML

GladeXML is the XML format used by the Glade Interface Designer. It creates forms that can then be used in conjunction with the libglade library using GTK+. Glade provides a graphical interface development environment in the model of Visual Studio, C++ Builder and so on [31].

B. XUL

XUL (XML User Interface Language), pronounced zool, is an user interface markup language developed to support Mozilla, Firefox and Mozilla Thunderbird. XUL reuses many existing standards and technologies, including CSS, JavaScript, DTD, RDF and XPCom [32].

C. Open Laszlo and Flash

Open laszlo is a free application based on a markup language named LZX. On the client-side, it produces Flash code which can be runs thanks to a plugin rather commonly installed, and it produces also DHTML code by Java script [33].

D. MXML

MXML is an XML markup language introduced by the Macromedia in 2004. Apart the design of user interface, it can also be used in conjunction with ActionScript to implement complex business logic. alternative from Macromedia to XUL and XAML is used by Web applications in conjunction with developments dedicated for the interaction with the browser. It has a development tool named Flex [34].

E. XAML

XAML is like XUL an XML-based language of description of interface language for silverlight. But unlike XUL it is at start intended by Microsoft to be a means of creating rich Internet applications. Just as XUL is recognized by Firefox, XAML is recognized by Internet Explorer, with the advantage of a larger number of users. It thus supplements the universal .NET platform, with a language of interface. It is provided natively with Vista and is used for the graphic interface of Vista [19].

Table 1: Shows The Comparison of Most Popular UI Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Developer</th>
<th>Runtime</th>
<th>Processing Type</th>
<th>Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GladeXML (GNOME)</td>
<td>1998</td>
<td>Use of Glade</td>
<td>Compiled</td>
<td>C, C++, C8</td>
</tr>
<tr>
<td>XUL (Mozilla)</td>
<td>1999</td>
<td>XUL Run</td>
<td>Interpreted</td>
<td>ECMA-Script, C++</td>
</tr>
<tr>
<td>Open Laszlo (Macromedia)</td>
<td>2003</td>
<td>XUL Player</td>
<td>Compiled</td>
<td>ActionScript</td>
</tr>
<tr>
<td>XAML (Microsoft)</td>
<td>2005</td>
<td>Win/X / Silverlight</td>
<td>Compiled</td>
<td>.NET languages, JavaScript</td>
</tr>
</tbody>
</table>
UI Test Automation Types:

1. For easy accessibility with low security the UI Test Automation will be easy.
2. For moderate accessibility with medium security the UI Test Automation will be moderate.
3. For high security the UI Test Automation will be very difficult.

So Silverlight UI Test Automation is very difficult because XAML accessibility is Difficult and has high security.

3. Types of UI Test Automation

Software test automation has evolved through several generations of tools and techniques. Test Automation is started from Character user Interface to till now. It can be classified based on testing strategy [12][24]. They are:

A. Capture/playback tools record the actions of a tester in a manual test, and allow tests to be run unattended for many hours each day, greatly increasing test productivity and eliminating the mind-numbing repetition of manual testing. However, even small changes to the software under test require that the test be recorded manually again [4][11]. Therefore this first generation of tools is not efficient or scalable [12][16].

B. Scripting is a form of programming in computer languages specifically developed for software test automation, alleviates many issues with capture/playback tools. However, the developers of these scripts must be highly technical and specialized programmers who work in isolation from the testers actually performing the tests. In addition, scripts are best suited for GUI testing and don’t lend themselves to embedded, batch, or other forms of systems. Finally, as changes to the software under test require complex changes to the associated automation scripts, maintenance of ever-larger libraries of automation scripts becomes an overwhelming challenge [4][11][12][17].

C. Data-driven testing is often considered separately as an important development in test automation [6]. This approach simply but powerfully separates the automation script from the data to be input and expected back from the software under test. This allows the data to be prepared by testers without relying on automation engineers, and vastly increases the possible variations and amounts of data that can be used in software testing. This breaking down of the problem into two pieces is very powerful. While this approach greatly extends the usefulness of scripted test automation, the huge maintenance chores required of the automation programming staff remain [11][13][16].

D. Keyword-based test automation breaks work down even farther, in an advanced, structured and elegant approach [6]. This reduces the cost and time of test design, automation, and execution by allowing all members of a testing team to focus on what they do best. Using this method, non-technical testers and business analysts can develop executable test automation using “keywords” that represent actions recognizable to end users, such as “login”, while automation engineers devote their energy to coding the low level steps that make up those actions, such as “click”, “find text box A in window B”, “enter User Name”, etc. Keyword-based test design can actually begin based on documents developed by business analysts or the marketing department, before the final details are known. As the test automation process proceeds, bottlenecks are removed and the expensive time of highly trained professionals is used effectively. Organizing test design and test automation with the keyword framework eliminates time [11][14].
E. UI Test Automation peer classes provide a powerful framework for organizing test design, test Automation and Test execution by AutomationID properties [20][21]. Using these testers can pass values to controls as well as they can invoke events for that controls, an automation engineer can focus on automating actions as individual building-blocks that can be combined in any order to design a test. Non-technical test engineers and business analysts can then define their tests as a [Test class], and execute their tests automatically without creating any additional code. Automation Peer test design takes place in a spreadsheet, with actions listed consecutively in a clear well organized sequence of Actions, test data and any necessary GUI interface information are stored in their own spreadsheets, from which they can be called by the main test module. Tests are then executed from right within the spreadsheet, using custom-built Automation Test classes. Automation Peer classes allow testing teams to create a much more effective test automation framework, overcoming the limitations of other methods. Automation Peer Testing significantly reduces the maintenance burden by allowing users to define their tests at the business process level. Rather than defining tests as a series of interactions with the UI, test designers can define tests as a series of business actions. It will be the job of the automation engineer to update the actions affected by the UI changes, and this update will only need to be made in only one place, rather than in multiple test scripts. Automation Peer Accessibility classes method is the best choice for Dot Net web based Silverlight Applications [11][20][25].

4. PERFORMANCE ANALYSIS

For any kind of software Applications Performance analysis is very important. From the performance analysis and comparative study one can easily choose their choice based on the analysis report. In this paper we took input data from like Text, Excel, XML, MS-SQL and Oracle. UI Automation Report will be produced in Text file, Excel sheet and XML. Silverlight is Microsoft’s new cross browser or delivering richer interactive applications to users over the web. Silverlight 2.0 is Microsoft’s second release of the Silverlight platform. Now a day’s web sites are developed with this Silverlight for best visualization and attraction.

A. Input Data

We maintained records in various databases like Text, Excel, XML, MS-SQL and Oracle. In each data bases we maintain the same records for this performance analysis. The UI Test Automation is initially started with 5000 records, then 10000, 15000 and 20000. Finally the average case is taken for our analysis.

1. Text File

Maintaining records in text file is very easy, but security is very less. Records are stored in a row and columns are separated by commas. File size is very small. Searching a particular record based some values are not so easy.

2. Excel sheet

In Excel, Data Entry is very easy and look and feel is very nice. Data records are stored in a row like a matrix. Excel provides little bit security when compared with Text file. Searching a value and sorting on a particular column are easy.
Comparative Performance analysis of UI Test Automation from various Databases for new web applications

3. XML Data

XML Files approach is Easy to use in any application. Data update is very simple and can be accessed via standard XML APIs. XML Data is universally understandable, portable and Best suitable for internet. Data retrieval is very fast. Now XML has advanced locking, synchronization and concurrency control mechanism. If we want ease of use and ability to transfer data easily between multiple applications than XML files approach seems right one [26][30].

- Web publishing: XML allows you to create interactive pages, allows the customer to customize those pages, and makes creating e-commerce applications more intuitive. With XML, you store the data once and then render that content for different viewers or devices based on style sheet processing using an XSL/XSLT processor.

- Web searching and automating Web tasks: XML defines the type of information contained in a document, making it easier to return useful results when searching the Web.

- General applications: XML provides a standard method to access information, making it easier for applications and devices of all kinds to use, store, transmit, and display data.

- E-business applications: XML implementations make electronic data interchange (EDI) more accessible for information interchange, business-to-business transactions, and business-to-consumer transactions.

- Metadata applications: XML is easier to express metadata (Unified Modeling Language design models or user interface properties, for example) in a portable, reusable format.

Pervasive computing: XML provides portable and structured information types for display on pervasive (wireless) computing devices such as PDAs, cellular phones, and others

```xml
<?xml version="1.0"?>
<xml-structure-describe customer details...>
  <customer>
    <record>
      <name>Appsmith</name>
      <address>106 Pt Calapex</address>
      <city>Pondicherry</city>
      <country>India</country>
      <phone>9786554175</phone>
      <email>appsmith_74@yahoo.com</email>
    </record>
    <record>
      <name>John</name>
      <address>12 Magna Rue</address>
      <city>Pondicherry</city>
      <country>India</country>
      <phone>9345197938</phone>
      <email>jonmtech@gmail.com</email>
    </record>
    <record>
      <name>Appsmith</name>
      <address>46 MGR Road</address>
      <city>Chennai</city>
      <country>India</country>
      <phone>9245821096</phone>
      <email>smith_2k@yahoo.com</email>
    </record>
  </customer>
</xml-structure-describe>
```

Figure 3 : XML Data For Input

4. MS-SQL

MS-SQL is a Microsoft server database with the Following Features:

- Security of data from unauthorized user.
- Compatible for all dot net Applications.
- Transparent Data Encryption. The ability to encrypt an entire database.
- Auditing. Monitoring of data access.
• Data Compression. Fact Table size reduction and improved performance.
• Resource Governor. Restrict users or groups from consuming high levels or resources.
• Hot Plug CPU. Add CPUs on the fly.
• Performance Studio. Collection of performance monitoring tools.
• Installation improvements. Disk images and service pack uninstall options.
• Dynamic Development. New ADO and Visual Studio options as well as Dot Net 3.
• Entity Data Services. Line Of Business (LOB) framework and Entity Query Language (eSQL)
• LINQ. Development query language for access multiple types of data such as SQL and XML.
• Data Synchronizing. Development of frequently disconnected applications.

5. ORACLE
Oracle has many advantages and features that makes it popular and thereby makes it as the world’s largest enterprise software company. Oracle comes with new versions with new features implemented in new version while the features of earlier versions still being maintained [28][29]. Some features of Oracle listed below.
• Role Based Security and fast adoption standards.
• Backward compatibility and Automatic Diagnostic Repository.
• Oracle is used for almost all large applications.
• Ease of customization and installation.

• Scalability and Performance (Concurrency, Consistency, Locking Mechanisms and Portability)
• Manageability (Self managing database, OEM, SQL*Plus, ASM, Scheduler, Resource Manager)
• High availability, Backup and Recovery
• Business Intelligence (Data Warehousing, OLAP, Data mining, Partitioning)
• Data integrity/Triggers

Information Integration Features (Distributes SQL, Oracle Streams)

B. UI Test Automation
UI Test Automation is very essential for Software industries to reduce test time, cost and man power. Now-a-days web applications are developed by new technologies like Silverlight, JAVA FX, FLEX, etc. Silverlight is new .NET technology to develop rich interactive Internet applications [11][19][22][23]. Testing these kinds of applications are not so easy to test, especially the User interface test automation is very difficult. Dot net 3.5 Framework Provides Automation Peer Accessibility class, using this we can develop test automation for Silverlight applications [1][3][11].

In this paper, we took five kinds of input data like Text, Excel, XML, MS-SQL and Oracle. For UI Test automation we took Automation Peer Accessibility classes method, this is the best choice for Silverlight Applications [20][25][27]. The output of UI Test Automanon is generated in Excel sheet with both input and output data.
UI Test Automation program will undergo slight modification if any controls changed, so common reusable code is maintained for all UI Test Automation [3][5][7][8]. The design of UI Test Automation with Input data types, Test Method types and Output report types is shown in figure 6 [10][19][20].

![Figure 6: UI Test Automation With Input Data Types, Test Method Types And Output Report Types.](image)

The sample code for UI Test automation using Automation peer method is given below [24][25][30]. Initially one process object is created and then we get the handle from the running browser. Tree walker object is created to find a particular element in the web page [20]. Using Silverlight objects we can set values to text box and then we click Buttons like add and update. These things are done by our program code test method [23][24][25].

```csharp
//Test.cs
[TestMethod]
public void TestMethod()
{
    TreeWalker tw1 = new TreeWalker(new PropertyCondition(AutomationElement.AutomationIdProperty, "TextBox");
    AutomationElement searchTextBox = tw1.FirstChild(browerInstance);
    TreeWalker tw2 = new TreeWalker(new PropertyCondition(AutomationElement.AutomationIdProperty, "Button");
    AutomationElement searchTextBox = tw2.FirstChild(browerInstance);
    SilverlightApp app = ActiveBrowser.SilverlightApp[0];
    app.FindElementByText("TextBox"); Text="Appase";
    app.FindElementByText("TextBox2"); Text="69-pk street Kolkata"
    app.FindElementByText("TextBox3"); Text="Pondicherry"
    app.FindElementByText("TextBox4"); Text="India"
    app.FindElementByText("TextBox5"); Text=976554175
    app.FindElementByText("TextBox6"); Text=appas_2g@yahoo.com
    app.FindElementByText("Add"); Use.Click();
}
```

![Figure 7: Sample Code For UI Test Automation](image)

Silverlight testing must collect information about all the controls available on that testing page then it form a tree with UI elements as nodes. The particular element is searched in Depth first searching technique. There should not any loop with in the search tree to search a particular element in search tree. Finally the value is passed to the particular element by UI Automation.

### C. Output Report

The output of UI Test Automaton can be easily updated in Excel sheet with both input and output data. The output report for UI Automation of Silverlight applications is as shown in the figure.

![Figure 8: Excel Output Report](image)
5. RESULTS AND DISCUSSIONS

The performance analysis is done in terms of security, reliability, portability, Integrity and processing time. We maintained records in various databases like Text, Excel, XML, MS-SQL, Oracle. In each data bases we maintain the same records for this performance analysis. The UI Test Automation is initially started with 5000 records, then 10000, 15000 and 20000. Finally the average case is taken for our analysis. The Approximately estimated time in nano Seconds for 5000, 10000, 15000 and 20000 records is given as shown in the table 1.

Table 1 : Different Database Records and Their UI Automation Processing Time

<table>
<thead>
<tr>
<th>Databases</th>
<th>No. of Records</th>
<th>Time (Sec)</th>
<th>Time for UI Automation (NS)</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text (Time in ms)</td>
<td>250</td>
<td>750</td>
<td>130</td>
<td>1750</td>
</tr>
<tr>
<td>Excel (Time in ms)</td>
<td>300</td>
<td>800</td>
<td>110</td>
<td>1800</td>
</tr>
<tr>
<td>XML (Time in ms)</td>
<td>100</td>
<td>600</td>
<td>140</td>
<td>1600</td>
</tr>
<tr>
<td>MSSQL (Time in ms)</td>
<td>400</td>
<td>900</td>
<td>1450</td>
<td>1900</td>
</tr>
<tr>
<td>Oracle (Time in ms)</td>
<td>450</td>
<td>950</td>
<td>1000</td>
<td>1950</td>
</tr>
</tbody>
</table>

The graph is plotted for the average case. In this Experiment the average number of records is taken as 12500 and the same records are maintained in all data files like text file, Excel sheet, XML Data, MS-SQL and Oracle table. The graph shows that the UI Test Automation time Using XML is very less when compared with others. The overall UI Test Automation processing time for each database is shown on the top of each column as shown in the bar chart.

6. CONCLUSION

There are several methods for UI automation, but AutomationPeer method is best for Silverlight applications. MS-SQL and Oracle Database provide more security, but it takes more access time when compared with XML data. Developer can choose any database according to their needs, but XML is best input data for Silverlight UI Automation, because entering data is easy and Transferring Time is also less. For Security purpose...
MS-SQL and Oracle are best. Without considering security, XML data is best input for Silverlight UI Automation. Automation Peer Method of UI Test Automation is best for Silverlight UI Test Automation.

To generate output report, Excel is best one when compared with others like Text, XML, MS-SQL and Oracle.

REFERENCES


[22] http://www.jeff.wilcox.name/2008/03/silverlight2-unit-testing

Author's Biography

Appasami. G received his Master of Science degree in Mathematics, Master of Computer Applications Degree and Master of Technology degree in Computer Science and Engineering from Pondicherry University, Pondicherry, India. Currently he is working as a lecturer in the Department of Computer Science and Engineering, Dr. Pauls Engineering College, Villupuram, Tamil Nadu, India. His Area of interests includes image processing and web technology.

Nakkeeran. R received his Master of Science, Master of Computer Applications Degree and Master of Engineering (Computer Science and Engineering) from Anna University, Chennai, India. Since 2007, he has been working as a Professor in the Department of Computer Science and Engineering, Dr. Pauls Engineering College, Villupuram, Tamil Nadu, India. His research interests includes web technology, DBMS and Software Engineering.

Suresh Joseph. K received his Bachelor of Engineering Degree from Bharathiyar University and Master of Engineering (Computer Science and Engineering) from University of Madras, Chennai, India, in 2002. Since 2006, he has been an Assistant Professor at the Department of Computer Science, Pondicherry University, and Pondicherry, India. His research interests includes Image processing and Software Engineering.
Improved Reliability In MANET’s Using Enhanced Double Coverage Broadcasting (EDCB)

Murali Mohan Reddy¹  R.Balakrishna²  U.Rajeswar Rao³  G.A.Ramachandra⁴

ABSTRACT

The broadcast operation, as a fundamental service in mobile adhoc networks (MANETs), is prone to the broadcast storm problem if forwarding nodes are not carefully designated. The objective of reducing broadcast redundancy while providing high delivery ratio under high transmission error rate is a major challenge in MANETs. This paper proposes a simple broadcast algorithm, called enhanced double-covered broadcast (EDCB), which takes advantage of broadcast redundancy to improve the delivery ratio in an environment that has rather high transmission error rate.

Keywords: Reliability, enhanced double coverage, MANETS, Performance evaluation, forwarding node

1. INTRODUCTION

A mobile adhoc network (MANET) enables wireless communications between participating mobile nodes without the assistance of any base station. The two nodes which are not in each other transmission range calls for a supporting intermediate node that relays the message and sets up communication. Further the broadcast operation place important fundamental role in the MANETs system by the way of radio transmission. With an added advantage of a single node transmitting packets, to the neighbors who can receive the message. But the MANETs system suffers from high transmission error rate owing to the high transmission contention and congestion. Hence providing higher reliability for broadcasting operation under dynamic MANET[16] system becomes an important concern and it posses a major challenge. And may be tackled by selecting an ideal error free fully covered environment with high transmission delivery ratio, single forwarding node which is entirely different from the existing algorithms that are based on probability. In that direction the acknowledgment messages (2ACKs)[16], used to ensure broadcast delivery ratio and E2ACK[17,16] fulfilling the requirements of receivers to send ACKs in response to the reception of a packet may becomes another bottleneck which rises to channel congestion and packet collision, are has to be effectively addressed. Hence the present paper aims at ad hoc networks using double coverage for improved broadcasting reliability.

2. ENHANCED DOUBLE COVERAGE BROADCAST (EDCB) – A NOVEL PROPOSAL

With the goal of reducing the number of forwarding nodes without sacrificing the broadcast delivery ratio, here with it is proposed with a simple broadcast algorithm designated as called enhanced double coverage broadcast with feature of broad casting redundancy and higher delivery ratio with lowered high transmission error rate. $G = (V, E)$: a unit disk graph as in Fig 1

Where $V$: Node set (set of wireless mobile nodes)
$E$: Edge set (set of bi-directional links between neighboring nodes)

Two nodes are considered as neighbors if and only if their geographic distance is less than the transmission range $r$.

In a localized broadcast protocol, a node $v$ is equipped with a $k$-hop subgraph $G_k(v)$ for a small $k$, such as $k = 2$ or 3. $G_k(v)$, induced from $k$-hop information of $v$, is $\{N_k(v), E_k(v)\}$. $N_k(v)$ denotes the $k$-hop neighbor set of node $v$ which includes all nodes within $k$ hops from $v$ (and also includes $v$ itself).

$H_k(v)$ denotes the $k$-hop node set of $v$ which includes all nodes that are exactly $k$ hops away from $v$; that is, $N_0(v) = H_0(v) = \emptyset$, $N_k(v) = N_{k-1}(v) \setminus H_k(v)$, $H_k(v) = N_k(v) \setminus N_{k-1}(v)$, for $k=1$. For convenience, 1-hop neighbor set $N_1(v)$ and 1-hop node set $H_1(v)$ are represented as $N(v)$ and $H(v)$, respectively. $E_k(v)$ denotes the set of links between $N_k(v)$, excluding those links between $H_k(v)$. That is, $E_k(v) = N_k(v) \setminus E_k(v)$. For example, if $v$ has 1-hop neighbor information, then it knows all its neighbors, but not the links between these neighbors. If $V$ is a node set, $N(V)$ is the union of the neighbor sets of every node in $V$, that is, $N(V) = \bigcup_{w \in V} N(w)$.

![Figure 1: Disk Graph of MANETS](image)

### 2.1 Algorithms

#### 2.1.1 Dynamic Neighbor-Designating Broadcast Algorithm

1. Let $u$ be the sender's broadcast packet, let it designate some neighboring node as its forwarding node set $F(u)$ to cover its 2 hop node set $H_2(u)$ thus $u$ sends the packet together with $F(u)$.

2. As the node $v$ receives the packet from $u$ for the first time, if $v$ is not designated as a forwarding node by $u$, it does nothing; otherwise it becomes a new sender and goes to step 1 above and is as shown in the Fig 2.

![Figure 2: Multiple Relays (MPR)](image)

The algorithms of the class of dynamic neighbor-designating broadcast algorithms adopting greedy strategy so as to designate minimum number of forwarding nodes are selected so that other neighbors can take the non-forwarding status.

In this proposal, multipoint relays [8, 9] are adopted and selected as the forwarding nodes to propagate link state messages. The MPRs are selected from 1-hop neighbors to cover 2-hop neighbors. Forwarded nodes are not considered for a node to select its MPRs and, therefore, the entire set of 2-hop neighbors must be covered (Fig.1). Specifically, $v$ selects its forwarding node set $F$ from all candidate neighbors $X = H(v) = N(v) \setminus \emptyset$ to cover its uncovered 2-hop neighbors $U = H_2(v) = N_2(v) \setminus N(v)$ with a simple greedy algorithm used in the set coverage problem [10]. These forwarding nodes set selection process (FNSSP) are detailed as follows.

#### 2.1.2 Forwarding Node Set Selection Process (FNSSP) (for node $v$)

1. Initially, $X = H(v)$, $U = H_2(v)$, and $F = \emptyset$.
2. Find $w$ (in $X$) with the maximum effective neighbor degree $\text{degree} (w) = |N(w) \setminus U|$.
3. $F = F \setminus \{w\}$, $U = U \setminus N(w)$, and $X = X \setminus \{w\}$.
4. Repeat steps 2 and 3 above until $U$ becomes empty as illustrated in Fig 3.

1612
2.1.3 Enhanced Double-Covered Broadcast Algorithm

Initiation

Broadcasting network requires a packet to receive by all nodes in the network. But transmission interference and the movement of the nodes may cause some nodes to lose the broadcast packet. The redundancy of the broadcast packet can bring more opportunities for a node to receive the packet successfully. Moreover, if the sender can retransmit the packet, the number of nodes that receive the broadcast packet is also increased.

The proposed enhanced double-covered broadcast (EDCB) algorithm [1,6] works as follows: When a sender broadcasts a packet, it selects a subset of 1-hop neighbors as its forwarding nodes to forward the packet based on a greedy approach. The selected forwarding nodes satisfy the following:

1. They cover all the sender’s 2-hop neighbors
2. The sender’s 1-hop neighbors are either forwarding nodes, or non-forwarding nodes but covered by at least two forwarding nodes.

After receiving a new broadcast packet, each forwarding node records the packet, computes its forwarding nodes and rebroadcasts the packet as a new sender. The retransmissions of the forwarding nodes are overhead by the sender as the acknowledgement of the reception of the packet. The non-forwarding 1-hop neighbors of the sender do not acknowledge the receipt of the broadcast.

The sender waits for a predefined duration to overhear the rebroadcast from its forwarding nodes. If the sender fails to detect all its forwarding nodes retransmitting during this duration [7,9], it assumes that a transmission failure has occurred for this broadcast. The sender then resends the packet until all the forwarding nodes’ retransmissions are detected or the maximum number of retries is reached. The sender may miss a retransmission from a forwarding node, and therefore resends the packet.

When the forwarding node receives a duplicated broadcast packet, it sends an ACK to acknowledge the sender.

The EDCB algorithm selects a set of forwarding nodes that form a virtual backbone of the network. The forwarding nodes are selected in such a way that they balance the average retransmission redundancy for the delivery of a broadcast packet throughout the entire network. The scheme avoids the broadcast storm problem: since only the forwarding nodes transmit the packet, the broadcast collision and congestion are reduced. This scheme also avoids the ACK implosion problem: the retransmissions of forwarding nodes are also used as the ACKs to the sender so that no extra ACKs are needed.

The failure of overhearing forwarding nodes’ relays will trigger the sender to retransmit the packet, so that the packet loss can be recovered in a local region. Each non-forwarding node is covered by at least two forwarding neighbors so that it can tolerate a single transmission error and its chance to receive the broadcast packet successfully is greatly increased even in a high transmission error rate environment. Moreover, the algorithm does not suffer the disadvantage of the receiver-initiated approach that needs a much longer delay to detect a missed packet.

Forwarding Node Set Selection Process

Assuming that each node \(v\) knows its 2-hop sub graph \(G_2(v) = (N_2(v); E_2(v))\). A forwarding node \(v\) uses the
FNSSP-DC algorithm to determine its forwarding node set $F(v)$: $v$ uses the FNSSP algorithm to find $F(v)$ in $H(v)$ to cover $N2(v)$ ($fvg$) (Fig. 2). Unlike the MPR algorithm [8] where only nodes in $H2(v)$ need to be covered by forwarding node set $F(v)$, the FNSSPDC algorithm guarantees that $v$’s 2-hop neighbor set $N2(v)$ (excluding $v$ itself) is completely covered by $v$’s forwarding node set $F(v)$. Since $v$ also transmits the packet to cover $H(v)$, any non-forwarding node in $H(v)$ is covered twice and is shown in Fig. 4.

1. Each node $v$ computes $X = H(v)$ and $U = N2(v)\backslash fvg$.
2. Node $v$ uses the FNSSP to find $F(v)$ in $X$ to cover $U$.

**Figure 4: Illustrations Of The Forwarding Node Set Selection Process Of The EDCB Algorithms**

The source of a broadcast operation uses the FNSSP-DC algorithm to determine its forwarding node set. Other forwarding nodes consider the impact of the sender of the broadcast packet. If $v$ is a designated forwarding node of $u$, that is, $v$ receives a new packet from $u$ and $v$ finds itself in $F(u)$, $v$ uses the FNSSPDC algorithm to determine its forwarding node set (Fig. 2): $v$ finds $F(v)$ in $H(v)\backslash N(u)$ to cover $N2(v)\backslash N(u)\backslash N(F(u)\backslash fvg)$. The goal of FNSSP-EDC is to cover all those nodes in the 2-hop neighborhood of $v$, excluding those that have been already covered by $u$ and those that will be covered by some other forwarding nodes of $u$.

**2.1.4 Forwarding Node Set Selection Process - Enhanced Double Coverage (FNSSP-EDC)**

Referring to Fig 1, a sample network where node 2 uses the FNSSP-EDC to select its forwarding nodes

1. Each node $v$ sets $X = H(v) \backslash N(u)$ and $U = N2(v) \backslash N(u) \backslash N(F(u)\backslash fvg)$.
2. Node $v$ uses the FNSSP to find $F(v)$ in $X$ to cover $U$.

**2.1.5 The Enhanced Double-Covered Broadcast (EDCB) Algorithm**

**Notations**

- $F(v)$: the forwarding node set of node $v$.
- $U(v)$: the uncovered 2-hop neighbour set of node $v$.
- $X(v)$: the selectable 1-hop neighbor set of node $v$.
- $P(u), F(u)$: unique broadcast packet $P$ forwarded by node $v$ that attaches vs. forwarding node set $F(v)$.
- $T_{wait}$: the predefined duration of a timer for a node to overhear the retransmission of its forwarding nodes.
- $R$: the maximum number of retries for a node.

**Proposed Algorithm**

1. When source $s$ wants to broadcast $P$, it uses the FNSSP-DC to find $F(s)$ and broadcasts $P(s; F(s))$.
2. When node $v$ receives $P(u; F(u))$ from $u$, $v$ records $P(u; F(u))$.
3. $v$ updates $X(v) = X(v) \backslash N(u)$ and $U(v) = U(v) \backslash N(u) \backslash N(F(u)\backslash fvg)$. If $v$ 2 $F(u)$ then if the packet has not been received before then $v$ uses the FNSSP-EDC to find $F(v)$ that covers $U(v)$ and broadcasts $P(v; F(v))$.
4. Else, $v$ sends an ACK to $u$ to confirm the reception of $P$ and drops the packet.

**end if**

3. When node $u$ has sent the packet, it starts a timer $T_{wait}$ and overhears the channel. After $T_{wait}$ is expired, if $u$ does not overhear all nodes in $F(u)$ to resend $P$ or to send ACKs, $u$ retransmits $P$ until the maximal number of retries $R$ is reached.
Improved Reliability in MANET's Using Enhanced Double Coverage Broadcasting (EDCB)

The above concept is supported by the following theorem, “Given a connected network, the EDCB algorithm works correctly based on the assumption that broadcasting through this network is an atomic operation”

Proof: We prove Theorem by contradiction. Assume that the network is not fully covered when broadcasting a packet with the EDCB algorithm, that is, we can find at least one node d such that d does not receive the broadcast packet from the source s. In Fig.5, the set C inside the circle represents the covered node set, C represents the uncovered node set. Therefore, s 2 C and d 2 C. Since the network is connected, there exists a path from s to d. Suppose node x is the uncovered node that is closest to s on the path, and v is the predecessor of x on the path. Based on the assumption, v has received the broadcast packet, say v has received the packet from node u for the first time. Because s 2 \( N(u) \), Theorem 1 guarantees that u covers s. This contradicts the assumption. Therefore, the EDCB algorithm guarantees the network is fully covered.

Figure 5: Illustration of the Proof of the Theorem

Working of the proposed EDCB Algorithm

1. When a node s starts a broadcast process, s uses the FNSSP-DC algorithm to select its forwarding node set \( F(s) \), and broadcasts the packet P together with \( F(s) \).

2. When a node v receives P from an upstream sender u, it records P. v also updates its \( X(v) = X(v) \cup N(u) \) and \( U(v) = U(v) \setminus N(u) \setminus N(F(u) \setminus \{v\}) \). Note that \( X(v) \) and \( U(v) \) are initialized to \( H(v) \) and \( H2(v) \).

Then, v checks whether it is a designated forwarding node of u. If not, v drops the packet and stops the process; otherwise, v further checks whether P is ever received. If P is a new packet for v, v uses the FNSSP-EDC algorithm to compute its forwarding nodes \( F(v) \) and sends P with \( F(v) \). If v has already received P from another node, v will not forward P, but send an ACK to u to confirm the reception so that u will not retransmit the same packet at a later time.

3. When the sender u broadcasts P, it waits for a predefined duration \( T_{wait} \) to overhead the retransmission of its forwarding nodes. If u overhears a retransmission packet from its forwarding node v, u regards this as an ACK from v. u may receive explicit ACKs from some of its forwarding nodes to confirm the reception. If u does not overhear all of its forwarding nodes when the timer expires, it assumes that the transmission failure has occurred for this packet. u then determines a new \( F(u) \) to cover the rest of the uncovered \( U(u) \) and resends the packet until the maximal number of retries \( R \) is reached.

The Fig. 6 shows working of EDCB

3. SIMULATION

In order to analyze the performance of the proposed algorithm, we run the simulation under the Glomosim simulator test bed with SMTP wireless extension. The simulator parameters are listed in Table 1: the network area is confined within 1000×1000 m². Each node in the network has a constant transmission range of 350 m. We use a two-ray ground reflection model as the radio propagation model. The MAC layer scheme follows the
IEEE 802.11 MAC[6,7,9] specification. We use the broadcast mode with no RTS/CTS/E2ACK mechanisms for all message transmissions, including HELLO, DATA and E2ACK[16,17] messages. Since transmission errors may occur when nodes send messages in real wireless channels, we assume a probability $P$ for each wireless channel to successfully transmit a message. The movement pattern of each node follows the random way-point model: each node moves to a randomly selected destination with a constant speed between 0 and the maximum speed $V_{\text{max}}$. When it reaches the destination, it stays there for a random period $T_s$ and starts moving to a new destination. The pause time $T_s$ is always 0 in our simulation. The network traffic load also affects the performance of the protocol; we change the value of constant-packet-rate CPR (packet per second) while each packet has a constant length of 64 bytes. A node may fail to receive a message because of a transmission error, a transmission collision or the node’s out-of-range movement. After sending a message, a node will wait for a period of time $T_{\text{wait}}$ and resend the message until it reaches the maximum value $R$. Each simulation was run for 100 seconds. In order to avoid the initialization bias of the system state on the broadcast operation, we first make all nodes move around within the area for 1000 seconds so that they can thoroughly exchange HELLO messages to build up 1-hop and 2-hop neighbor sets. Then, some randomly selected nodes start to send broadcast packets. This procedure lasts for 100 seconds. To make sure all the broadcast packets propagate throughout the network, the simulation will last for another 10 seconds after the last broadcast process has been sent. We run the simulation 10 times to achieve a 90% confidence interval for the results.

### Table 1: Simulator Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulator</td>
<td>Advent</td>
</tr>
<tr>
<td>Network area</td>
<td>1000 X 1000</td>
</tr>
<tr>
<td>Transmission Rang</td>
<td>350m</td>
</tr>
<tr>
<td>MAC Layer</td>
<td>IEEE 802.11</td>
</tr>
<tr>
<td>Data Packet Size</td>
<td>64 Bit</td>
</tr>
<tr>
<td>Band width</td>
<td>4 Mb/s</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>90 s</td>
</tr>
<tr>
<td>Number Trials</td>
<td>10</td>
</tr>
<tr>
<td>Confidence Intervals</td>
<td>93 %</td>
</tr>
</tbody>
</table>

4. RESULT ANALYSIS

In this 1000x1000 network area at low mobility where $V_{\text{max}}$ is 1 meter per second (m/s) and low transmission error rate ($P_{\text{err}} = 0.8\%$). For data traffic load CPR at 10 packets per second (pkt/s), the hello interval HELLO is 1 second (s), and the waiting time $T_{\text{wait}}$ is 50 millisecond (ms), the identified effect of network size is $n$ to each metric and the network under this environment can be considered being static error-free. It is observed for the above conditions all algorithms have good delivery ratios (> 93%). Further the delivery ratio proposed EDCB-SD is greater than the other two algorithms namely DCB-ST, DCB-RE for all the ranges considered. Where as the delivery ratios of all DCB algorithms are slightly higher than AHBP-EX at dense ($n=1000$ network), which suggesting the EDCB-SD algorithm delivery ratio is superior benefiting from the retransmission mechanism.

The Fig.7-10 are presented with the performance analysis of the Proposed EDCB-SD algorithm. Fig. 7 shows the packet delivery ratio for the different nodes. From fig it can be observed that with increased nodes the packet delivery ratio increases. Further form the nature of the curve it may be depicted that the packet delivery ratio increases with increased number of nodes.
Improved Reliability in MANET’s Using Enhanced Double Coverage Broadcasting (EDCB)

Figure 7: Variations Delivery Ratio with Increased Number of Nodes

Figure 8 shows the Broadcasting ratio for the different nodes. From fig it can be observed that with increased nodes the broadcasting delivery ratio increases. Further from the nature of the curve it may be depicted that the broad casting delivery ratio increases with increased number of nodes.

Figure 8: Variations Broad Casting Ratio with Increased Number of Nodes

Fig.9 shows the broadcasting overhead for the different nodes. From fig it can be observed that with increased nodes the broad casting overhead increases. Further from the nature of the curve it may be depicted that the broad casting overhead decreases with increased number of nodes.

Figure 9: Variations Overhead With Increased Number of Nodes

Fig. 10 shows the Broad Casting end to end delay for the different nodes. From fig it can be observed that with increased nodes the broadcasting end to end delay increases. Further from the nature of the curve it may be depicted that the broad casting delivery ratio increases with increased number of nodes.

Figure 10: Variations End to End Delay with Increased Number of Nodes

From this simulation, we can see that all algorithms have high delivery ratios under the 1000x1000 that the network is almost static and transmission errorfree. DCBs and AHBP-EX have comparable performance under this scenario. Also, we notice that EDCB-SD performs best among three DCB algorithms

5. IMPLEMENTATION

The steps followed in the implementation of proposed EDCB-SD algorithms are shown below.
7. CONCLUSIONS

Algorithm provides high delivery ratio while suppressing broadcast redundancy. This is achieved by only requiring some selected forwarding nodes among the sender’s 2-hop neighbor set to forward the packet. The double-covered forwarding node set selection process provides some redundancy to increase the delivery ratio for non-forwarding nodes so that retransmissions can be remarkably suppressed when transmission errors are present.

6. ADVANTAGES OF EDCB AND COMPARISONS

The advantages of proposed EDCB are as follows:

1. Data redundancy is avoided, so that each node is promised to receive the packet only from one node.
2. The total delay for getting acknowledgement at source is reduced since data redundancy is removed.
3. In case of node failure, the node topology is rearranged so that child nodes of failed node receive the packets.

The comparison among various algorithms shown in Table 2 supports the proposed EDCB-SD algorithm:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDCB-SD</td>
<td>Forwarding nodes</td>
</tr>
<tr>
<td>DCB-ST</td>
<td>Forwarding nodes</td>
</tr>
<tr>
<td>DCB-RE</td>
<td>Forwarding nodes</td>
</tr>
</tbody>
</table>

7. CONCLUSIONS

Algorithm provides high delivery ratio while suppressing broadcast redundancy. This is achieved by only requiring some selected forwarding nodes among the sender’s 2-hop neighbor set to forward the packet. The double-covered forwarding node set selection process provides some redundancy to increase the delivery ratio for non-forwarding nodes so that retransmissions can be remarkably suppressed when transmission errors are present.
Improved Reliability in MANET’s Using Enhanced Double Coverage Broadcasting (EDCB)

considered. The simulation results show that the enhanced double-covered broadcast algorithm has

• high delivery ratio,
• low forwarding ratio,
• low overhead and low end-to-end delay

for a broadcast operation under high transmission error ratio environment. From the simulation, we observe that the EDCB is sensitive to the node’s mobility.

The EDCB provides full reliability for all forwarding nodes but not for non-forwarding nodes. In order to provide full reliability for all non-forwarding nodes, we can use the N-ACK mechanism such that a non-forwarding node will send a N-ACK message when the node notices a packet loss during the continuous broadcasting transmissions.

REFERENCES


Author’s Biography

M. Murali Mohan Reddy is a research scholar in the department of Computer Science and Technology, Sri Krishnadevaraya University, Anantapur, Andhra Pradesh, India. His research focuses on mobile ad-hoc networks (MANETs) simulation, modeling and broadcast applications. His other research areas are improving Data Warehouse performance through efficient maintenance, privacy-preserving Data Network Security. He has published 5 research articles in National and International Journals and symposia. He is the Life member of IAENG.

R. Balakrishna is a research scholar in the department of Computer Science and Technology, Sri Krishnadevaraya University, Anantapur, Andhra Pradesh, India. His main research Interests are Wireless mobile ad-hoc networks simulations, Sensor Networks, Artificial Neural Networks, Data Mining,
Improved Reliability in MANET's Using Enhanced Double Coverage Broadcasting (EDCB)

Operating Systems and Security. He has published 26 research articles in National and International Journals and attended Conferences across India and abroad. He is the Life member of Indian Society for Technical Education (LMISTE), ACM, DELNET and IAENG.

Dr. U.Rajeswar Rao obtained his Ph.D., from Sri Krishnadevaraya University, Anantapur, Andhra Pradesh, India. He is currently working as Professor in the Sri Krishnadevaraya University. He has attended and organized number of national and international conferences and symposia in the field and guided about 25 Ph.D students. He has published about 50 research articles in National and International Journals in the area of Mathematics & Computers Science.

Dr. G.A.Ramachandra obtained his Ph.D., from Sri Krishnadevaraya University, Anantapur, Andhra Pradesh, India. He is currently working as Associate Professor and Head of the Department of Computer Science & Technology, Sri Krishnadevaraya University. He has published numerous conference papers, National and International Journal articles in the area of Data Mining, Ad hoc networks, Neural Networks and Networking Security. He has been the thesis supervisor of more than 10 research scholars.
An Efficient Dynamic Router Approach to Defeat DDOS Attacks

S S Nagamuthu Krishnan1 V. Saravanan2

1 Ph.D. Research Scholars, Assistant Professor, Thiagarajar School of Management, Madurai, Bhartathiar University, Coimbatore – 641 146, Tamilnadu, India, Email: ssnkrishnan@gmail.com
2Professor & HOD, Department of Computer Application, Karunya University, Coimbatore – 641 114, Tamilnadu, India. Email: tvsaran@hotmail.com

ABSTRACT
Denial-of-service attacks represent a major threat to modern organizations that are increasingly dependent on the integrity of their computer networks. Recently many prominent web sites face so called Distributed Denial of Service Attacks (DDoS). Even there are many approaches to avoid DDOS attacks no approaches completely satisfies the protection yet.

A new approach to combating such threats introduces dynamic routers into the network architecture. These dynamic routers offer the combined benefits of intrusion detection, and work collaboratively to provide a distributed defense mechanism. The paper provides a detailed description of the design and operation of the algorithms used by the dynamic routers and demonstrates how this approach is able to defeat the attacks. It is proposed that the adoption of a dynamic router approach in protecting networks overcomes many of these weaknesses and therefore offers enhanced protection.

This paper discusses a simple, effective, and straightforward method for using ingress traffic filtering to prohibit DDoS attacks which use forged IP addresses to be propagate from 'behind a nearest router aggregation point.

Keywords: DDoS Attack, Time slot, Transfer Rate, Signature of a Packet, Security, Incentive, Economical.

1. INTRODUCTION
Internet-enabled business, or ebusiness, has mushroomed into a significant part of the US economy, yet further advancement of e-business is plagued by various Quality-of-Service (QoS) and security problems. One of the worst is the Distributed Denial-of-Service (DDoS) attack, which aggregates junk data traffic from up to thousands of computers into a formidable volume and floods and effectively blocks a certain victim website. DDoS attacks have drawn a lot of media attention since the landmark attacks on a large portfolio of famous e-business websites including Yahoo!, Amazon, CNN, eBay, and E*Trade, in early 2000 [Klein bard 2000]. Cavusoglu et al. [2002] estimate that the firms involved lost more than 2.8% of their market capitalization. Academic discussion also quickly followed up with proposals that can be broadly classified into two categories: technological solutions [Wang and Reiter 2004; Badishi et al. 2004; Xiang et al. 2004; Mirkovic et al. 2005 Chapter 7], and economic solutions [Geng and Whinston 2000; Geng et al. 2002].

Figure 1 illustrates the mechanisms of a DDoS attack. There are two separate stages of DDoS attacks: recruiting zombies and flooding the victim [Chang 2002]. In the recruiting stage (steps 1 and 2), security flaws are used to break into master computers and a large set of zombie computers is established. In the flooding stage, a direct attack or a reflector attack is launched and synchronized traffic with IP spoofing [Geng and Whinston 2000] disables the services of the victim (steps 3 and 4).
An Efficient Dynamic Router Approach to defeat DDOS attacks

Figure 1: The Mechanism of DDOS attacks
It is now well-understood that several cooperative technological solutions including cooperative filtering and cooperative traffic smoothing by caching (as we will shortly discuss) will be quite effective against DDoS attacks but some of those cooperative technological solutions were proposed as early as in 2000 (e.g. RFC 2827—ubiquitous ingress filtering), they are clearly not effectively deployed.

But the drawback of this method is that, the traffic is found throughout the path from source to destination. Hence the bandwidth of the network is not utilized properly and also affects economy of ISPs and ICPs. Since there is traffic the service for the request is delayed for long time. So here we propose a new solution called Dynamic router approach which works based on the conditions—Ip-address, request parameters. In order to identify the attackers’ request first Ip address of the source is verified whether it is from a known source or not. Then the parameter types are compared. When the request is found to be the attackers request then the request will be blocked.

2. The Cooperative Technological Solutions To DDoS Attacks
2.1 Cooperative Filtering
Cooperative filtering is the first cooperative technological solution. Cooperative filtering works in three steps: alarming, tracing, and filtering (illustrated in Figure 2). By analyzing the pattern of network traffic, Intrusion Detection Systems (IDS) identify suspicious traffic and send out alarms. Following the alarms, a tracing mechanism kicks in to track back each attack path as far as possible. Finally, a series of filters along every attack path are configured to filter out attack traffic. In the best scenario, a tracing mechanism may find the computers (zombies) that are initiating attack traffic, and may inform the responsible ISPs to take them offline.

Figure 2: The Process of Co-operative Filtering
2.2 Cooperative Caching
Cooperative caching is an effective solution to DDoS attacks when cooperative filtering is costly to implement, or when attack traffic is well concealed in legitimate data requests such that pattern recognition is technically difficult. Cooperative caching and filtering can also be jointly deployed so that attack traffic is both reduced and diverted, resulting in a more effective defense. One important technological issue in using cooperative caching to defeat DDoS attacks is the fact that only relatively static content can be cached. If a DDoS attack targets dynamic content or protocols (such as ICMP ECHO, SYN floods, BGP floods), and traditional caching solutions cannot divert it. This issue is now partially addressed in two ways. First, standards like Edge
Side Include (ESI, see www.esi.org) enable caching of dynamic content. Second, more ISPs start screening and restricting control packets. For example, the attacks using ping commands are no longer effective when ICMP traffic is restricted.

2.3 The Broken Incentive Chain

Despite the fact that cooperative filtering and cooperative caching are two effective technological solutions against DDoS attacks, to date they have rarely been deployed on the Internet because of incentive chain.

There are two major sources driving the flow of digital content on the commercial Internet: end users’ demand to consume digital content and ICP’s demand to publish digital content. As shown in Figure 3, while both end users and ICP’s only pay directly to their ISPs for Internet connections, those regional ISPs in turn pay larger regional ISPs and backbone ISPs for the connectivity to the core of the Internet. We call this series of payments the “incentive chain,” which acts as glue to link all parties together in the end-to-end transmission of digital content.

2.4 Lack Of Incremental Payment Structure And The Failure Of Co-operative Filtering

One important implication of this conservative practice in uplink planning is that most of the time ISPs have abundant unused residue bandwidth that they have already paid for to the upper-level ISPs. ISPs are willing to provide such an unused resource for better consumer retention, and on the surface it appears not to hurt anybody else. However, it actually leads to devastating consequences on cooperative filtering against DDoS attacks, once we look at the question: what are the costs and benefits for an ISP to engage in cooperative filtering? While the cost side includes the administrative work in setting up and maintaining filters, and the reduction of transmission performance due to filtering overhead; the benefit side often includes little to nothing as long as DDoS attacks only consume some of the residue bandwidth, which is unused anyway.

The inability of victims in DDoS attacks to motivate ISPs who are in the best position to filter attack traffic is the direct result of the lack of incremental payment structures on the Internet. By selling and buying Internet access on a subscription basis, ISPs have little incentive to control traffic volumes as long as it does not create congestion in their own neighborhoods, simply because the marginal cost for transmitting additional data packets is zero. Additional bandwidth may be used to initiate DDoS attacks and harm ICPs far away. However, this does not provide any incentive for local ISPs to take any action. Clearly, when it comes to a DDoS attack, the incentive chain is broken.

2.5 Caches On The Edge Of The Internet: Inaccessible Treasures

The optimization of an incentive chain is all about the tradeoffs between the costs and benefits of various possible incentive schemes. As we noted before, cooperative filtering is actually costly to ISPs because of administrative costs and performance reduction. Alternatively, if DDoS attack traffic can be diverted to a lot of cache servers through cooperative caching, it can
be an effective solution, as it prevents the accumulation of traffic from happening. Since cache servers already exist, as long as cooperative caching only uses redundant cache capacity, it incurs little cost to any party involved, and thus is more cost efficient than cooperative filtering. Nevertheless, as shown in Figure 3, ISPs’ caches only serve their local users who pay for connections. Congestion at the ICP’s website does not provide any payment for cache servers on the demand side to engage in cooperative caching. Therefore, the resource is inactive for defending DDoS attacks, and again the incentive chain is broken.

3. EXISTING SOLUTION

3.1 Fixing The Incentive Chain Capacity Provision Network

A Capacity Provision Network (CPN), which would be a network of cache servers owned, operated and coordinated through capacity trading by different ISPs. A CPN is initially proposed for demand-side cache trading, the usefulness of which is supported by the fact that there exist positive network externalities across individual ISPs who provide caching services to their respective local users: when some ISPs are experiencing high demand for caching, other ISPs’ cache capacity may be idling. Therefore, by sharing the idling cache capacity with busy ISPs, total welfare increases. Cache trading is operated in a CPN market, which is organized by a market owner. We propose that the owner of the CPN market fits well in the intermediary’s role as we described it: the owner specializes in dealing with large numbers of ISPs who own cache servers, and the owner is a single entity that can deal with outside organizations on behalf of its participating ISPs. Figure 4 illustrates an incentive chain for CPN owner-intermediated cooperative caching. An ICP initiates the incentive chain by contracting with and paying the CPN owner for cooperative caching against any possible DDoS attacks. When a DDoS attack happens, the CPN owner decides which cache server is in the best position to dilute the traffic and then pays relevant ISPs to start cooperative caching, which completes the incentive chain. Of course, how much the ICP pays the CPN owner depends in turn on how much the owner pays ISPs.

4. PROPOSED SOLUTION

Figure 4 : Capacity Provision Network

The attacks are generally conducted by sending packets to the victim at a higher rate than they can be served, causing the Denial of legitimate service requests. In distributed Denial of service (DDoS) attacks, the aggregate traffic from several different sources is responsible for disabling the services provided by the victim.

The limitation of IP trace back problem is that identified machines might not be the actual attack sources. In fact only Zombies may be recognized and, therefore, more sophisticated schemes are required to locate the true origin of the attack.

Before seeing the solution in detail the things to be kept in mind are the maximum request size is 2GB, maximum number fields to be in the request is 100 (normally the users use only up to 20 fields).

In this solution using header information of packets - size of packets and number of fields and IP address we are
able to find out the attack in the first router itself. It avoids more traffic which penetrates through network. So this approach will save the time, and the network bandwidth is utilized properly. “Time Slot” is maintained for each request. The nearest router will check each request to see whether the request is legitimate or not. If there is more number of requests immediately one after other with packet size nearer to maximum then it is considered to be attack request and the request is dropped. The number of request arrived from unique IP address corresponding to particular router can be found from the router’s database. If the request is attack request, the transfer rate of the

Figure 5: Restricting Fake Packet

network is very low and the network bandwidth is blocked. Hence the service for the legitimate users is also denied. If the illegitimate request is identified in the first router itself, there is no need to find the source of the attack. Trace back is not at all needed.

In the example above, the attacker resides within 9.0.0.0/8. An input traffic filter on the ingress (input) link of “router 2”, which provides connectivity to the attacker’s network, restricts traffic to allow only traffic originating from source addresses within the 9.0.0.0/8 prefix, and prohibits an attacker from using “invalid” source addresses which reside outside of this prefix range. In other words, the ingress filter on “router 2” above would check. If the address also would have been same then the packet signature such as request size, the number of fields in each packet will be checked. If it is an attacker’s request then it will be blocked and so the traffic will be reduced.

Sample Data and Results

<table>
<thead>
<tr>
<th>Nodes in the Network</th>
<th>Time Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.078125</td>
</tr>
<tr>
<td>200</td>
<td>0.109375</td>
</tr>
<tr>
<td>300</td>
<td>0.109375</td>
</tr>
<tr>
<td>400</td>
<td>0.15625</td>
</tr>
<tr>
<td>500</td>
<td>0.15625</td>
</tr>
<tr>
<td>600</td>
<td>0.15625</td>
</tr>
<tr>
<td>700</td>
<td>0.171875</td>
</tr>
<tr>
<td>800</td>
<td>0.234375</td>
</tr>
<tr>
<td>900</td>
<td>0.234375</td>
</tr>
<tr>
<td>1000</td>
<td>0.265625</td>
</tr>
</tbody>
</table>

[YUN HUANG, 2007]The above graph depicts the CPN method. In that, Time taken for reaching the victim based on the network path is shown for the sample data.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Time taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.078125</td>
</tr>
<tr>
<td>200</td>
<td>0.078125</td>
</tr>
<tr>
<td>300</td>
<td>0.5</td>
</tr>
<tr>
<td>400</td>
<td>0.078125</td>
</tr>
<tr>
<td>500</td>
<td>0.078125</td>
</tr>
<tr>
<td>600</td>
<td>0.078125</td>
</tr>
</tbody>
</table>
An Efficient Dynamic Router Approach to defeat DDOS attacks

In our proposed solution the time taken to reach the victim is avoided. Since the attack is identified in the first router itself.

<table>
<thead>
<tr>
<th>No of Packets(No’s)</th>
<th>Transfer Rates (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>200</td>
<td>96</td>
</tr>
<tr>
<td>300</td>
<td>84</td>
</tr>
<tr>
<td>400</td>
<td>77</td>
</tr>
<tr>
<td>500</td>
<td>55</td>
</tr>
<tr>
<td>200</td>
<td>90</td>
</tr>
<tr>
<td>210</td>
<td>96</td>
</tr>
<tr>
<td>220</td>
<td>94</td>
</tr>
<tr>
<td>215</td>
<td>98</td>
</tr>
<tr>
<td>225</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 7

When compared to CPN, this method is cost effective because in CPN the attack request is not identified just when there is more network traffic the traffic is spitted and distributed to the near cache server to serve the request. In our solution, the request whether is attack request or a legitimate one is identified in the nearest router (first router the request passes through from the sender) and only the attacker’s LAN traffic gets affected and not the traffic through out the network is avoided. Hence the cache can be used efficiently to provide service for the legitimate requests.

5. CONCLUSION

Denial-of-services attacks can cause significant damage to web service providers. Currently, the Internet routing infrastructure does not provide means of locating the attacker nor avoiding such attacks. The rapid growth of denial-of-services attacks has led to a great number of proposed solutions. All the previously proposed methods concentrated mostly on determining the attack path. However, with our proposed solution we can easily safeguard any network from attack. While implementing this within LAN congestion may occur. Additional functions should be considered for future platform implementations, such as the implementation of multiple cache servers on the network in order to avoid the congestion with in the network facilitated by some complex congestion control algorithm.

REFERENCES


[10] Defeating DoS Attacks with IP Trace back Rafael P.Laufer* - Pedro B. Velloso** - Otto Carlos M. B. Duarte 2005*


Author’s Biography

Mr. S S Nagamuthu Krishnan has got his Bachelor’s degree in Physics from Madurai Kamaraj University during the year 1995 and MCA degree from Bharathiar University during the year 1998. He has obtained his MPhil in Computer Science from Bharathiar Universty in the year 2007. He has got more than 9 years of academic experience. His areas of interest are Object Oriented Analysis and Design, Computer Security, Data Structures & algorithms and Networking. He is also pursuing his research leading to Ph. D. in Network Security.

Dr. V Saravanan obtained his Bachelor’s degree in Mathematics from University of Madras during 1996 and Masters Degree in Computer Applications from Bharathiar University during 1999, and his PhD in Computer Science in the Computer Science and Engineering, Bharathiar University during 2004. He specialized on automated and unified data mining using intelligent agents. His research area includes data warehousing and mining, software agents and cognitive systems. He has presented many research papers in National, International conferences and Journals and also guiding 3 researchers leading to their PhD degree. He has totally 10 years experience in teaching including 3 years as researcher in Bharathiar University. He is the member of Computer Society of India, Indian Association of Research in Computing Sciences and many professional bodies.
Blocking Parameter Driven Virtual Topology Reconfiguration for IP-Over-WDM Networks with QoS Parameters

Ramasamy Mariappan 1
R.Shanmugalakshmi 2

1Department of Information Technology, Vinayaka Mission’s Aarupadai Veedu Inst. of Tech, Paiyanoor–603 104, Tamilnadu, INDIA. Tel : 09444812756, Email : mrmrama2004@rediffmail.com

2Assistant Professor, Department of Computer Science & Engineering, Government College of Technology, Coimbatore – 641 013, Tamilnadu, INDIA.

Abstract

Internet Protocol (IP) over Wavelength Division Multiplexed (WDM) networking technology with Terabits per second bandwidth become a natural choice for the future generation internet networks (Optical Internets), wide area network (WAN) environments and backbone networks due to its potential ability to meet rising demands of high bandwidth and low latency communication. Recently Virtual Topology Reconfiguration of IP over WDM networks has received greater attention among researchers. In this paper, we have presented a new approach for reconfiguring virtual topology of IP over WDM networks driven by blocking parameter with Quality of Service (QoS) requirements. The simulation results show that this new approach achieves better QoS performance in terms of blocking probability, throughput and latency.

Keywords: Blocking Probability, Lightpaths, Logical Topology, Virtual Topology Reconfiguration, QoS.

1. Introduction

In the last decade, WDM Optical fiber networking technologies brought a revolution in high-speed communication networks, which are now able to meet the high bandwidth demands of current voice and data traffic. The demand for bandwidth is growing at a rapid pace, and the Internet data traffic is expected to dominate voice traffic in the near future. With the IP playing a dominant role in wide area networking technology and advancements in wavelength routed WDM technology to provide enormous bandwidth, the IP over WDM networks [1] become the right choice for the next generation Internet networks.

The physical topology [2] consists of optical WDM routers interconnected by point-to-point fiber links and nodes in an arbitrary topology. In these types of networks, data transfer carried from one node to another node using lightpaths. A lightpath [3] is an all-optical path established between two nodes in the network by the allocation of same wavelength on all links of the path. In IP over WDM networks, lightpaths are established between IP routers.

A virtual topology [4] is a set of pre-established lightpaths established to provide all optical connectivity between nodes for a given traffic demand. The virtual topology is established logically through lightpaths, each identified by an independent wavelength, which provides end-to-end connectivity for transmission over the optical medium. The embedding of virtual topology over a physical topology results in minimizing the number of nodes that were actively involved in network transmission.
The virtual topology designed initially for a particular traffic may not be optimal for the changing traffic. The virtual topology designed over IP may need to be changed in response to changing traffic demands or due to failure of network elements. This process of changing the current virtual topology to a new one to adapt the dynamic change of traffic or failure of network elements is called Virtual Topology Reconfiguration [4].

Reconfiguration [5] is one of the significant characteristics of WDM optical network. The reconfiguration [6] is achieved by the fact that WDM optical networks provide an architecture in which logical connections can be embedded over the underlying physical connections. Reconfiguration [7] can be viewed as a process for providing a tradeoff between the objective function value and the number of changes to the current virtual topology. The objective function value decides how best the topology is suited for the changed traffic demand. Since the VTR problem is computationally intractable, heuristic solutions are desired to yield near optimal solution. There are two different approaches [8] of virtual topology reconfiguration to handle the dynamic traffic. In the first method, a new virtual topology is designed for each change in traffic. In the second method, reconfiguration is done with the objective of minimizing the objective function value and the number of light path changes.

This paper is organized as follows. A survey of the related work available in the literature is made in the section 2. Section 3 describes our proposed work including the QoS parameters. In section 4, the traffic model is described and section 5 and 6 present the heuristic algorithm and simulation results respectively. Finally, section 7 concludes the paper.

2. RELATED WORK

Many researchers have extensively studied about optical network and its reconfiguration problem for WDM network. Virtual topology design problem for WDM mesh network with the objective of minimizing average packet delay is given [1]. Linear programming methods for Virtual topology design problem with the objective of minimizing network congestion is proposed [2] [3]. Linear programming for the virtual topology design problem becomes computationally intractable; therefore heuristic approaches are made use of. Wavelength continuity constraint for the virtual topology design problem has been considered [2], i.e. it is assumed that nodes are not equipped with wavelength converters. Therefore it becomes necessary that a light path use the same wavelength on all the physical links. An extensive survey of virtual topology design algorithms has been carried out [4]. Linear programming and heuristic methods for different topologies are described and compared. Routing and wavelength problem and reconfiguration of virtual topology have also been dealt with. The combined problem of physical topology and virtual topology design has been taken up [5] using genetic algorithm.

There are many research work previously done in the reconfiguration of WDM networks. Reconfiguration of virtual topology for dynamic traffic is carried out with the aim of minimizing one or more objective functions, in order to maximize resource utilization. An integer linear programming method and resource budget for virtual topology reconfiguration problem with the objective of minimizing average hop distance has been proposed [6].
The authors also discuss on reconfiguration aimed at minimizing the number of lightpath changes for a mesh network. The dynamic reconfiguration for optical networks with tradeoff between optimality of the network and network disruption is described in [7]. The reconfiguration problem for wireless optical network is described in [8]. An adaptive mechanism without prior knowledge of the future traffic pattern is proposed [9]. Here the authors consider a slowly varying traffic pattern and consider addition or deletion of one lightpath at a time. A higher and lower watermark level is used to find when to reconfigure the network by adding or deleting the lightpaths.

A two-stage approach of reconfiguration with objective of minimizing average weighted hop count with a tradeoff between the objective function value and the number of changes to the virtual topology is considered [10]. The first stage is reconfiguration stage and the second stage is an optimization stage, which reduces the deviation from the optimal objective function value. Integrated services, differentiated services and multi protocol label switching (MPLS) have been discussed and the various possible service classes are also detailed. In differentiated services, packets are marked differently based on the service requirements. Constraint based routing enables determination of routes based on constraints like bandwidth or delay.

The dynamic reconfiguration problem with priority based addition and deletion of light path was presented in [11]. The QoS aware routing and trafficgrooming problem was described in [12]. The dynamic reconfiguration [13] of virtual topology requires a lot of control overhead and results in network disruption. In the present day WDM networks, a typical reconfiguration process in the order of tens of milliseconds corresponds to tens of megabits of traffic that must be buffered or rerouted at each node that is being reconfigured. If this disruption is not taken care properly, it will result in severe congestion and heavy data loss in the network as the traffic on the light paths is order of gigabits per second. The traffic generated belongs to various QoS classes and the service requirement for each QoS class is different. Hence the QoS parameters for the changing traffic to be taken into consideration while reconfiguring the virtual topology for an IP over WDM network.

### 3. Proposed Work

In the literature, simple reconfiguration algorithms were given for optical network [7],[8] without considering QoS parameters. The VTR algorithms [4],[10],[11] available are yielding minimum congestion. These all affect the performance of the IP-over-WDM networks, in particular Quality of Service (QoS) [12][13]. In this paper, reconfiguration algorithm for IP/WDM network driven by blocking parameter under dynamically changing traffic is proposed with QoS parameters namely, minimum delay, minimum blocking probability and maximum throughput.

#### 3.1. Problem Statement

The reconfiguration of virtual topology for optical network using IP over WDM technology is usually handled by modeling it as a Mixed Integer Linear Programming [13] or optimization problem which minimizes the objective function such as Average Weighted Hop Count of the Virtual Topology (AWHT), Congestion, number of lightpath changes, etc. There is a trade off between problem objectives and QoS parameters like, Blocking probability, Throughput, Delay, etc. In our research work we propose a new heuristic of blocking parameter driven reconfiguration of virtual topology for IP over WDM networks for dynamic traffic changes considering QoS parameters stated above. In this paper,
the VTR problem is formulated as an optimization problem of minimizing network congestion with improvising QoS. The total traffic on the virtual link from node i to j is given by,

$$T_{ij} = \sum_{s,d} T_{ij}^{s,d}$$  \hspace{1cm} (1)

where $T_{ij}^{s,d}$ gives the traffic from node s to d that employs the virtual link i to j. Congestion is defined as the maximum traffic flow in a light path due to all source destination node pairs and is given by,

$$T_{\text{max}} = \max_{i,j} T_{ij}$$  \hspace{1cm} (2)

3.2. Network Model
We consider a network [6] of N nodes connected by bi-directional optical links forming an arbitrary physical topology. Each optical link supports w wavelengths, and each node is assumed to have T transmitters and R receivers. We assume that each node is equipped with an optical cross connect (OXC) with full wavelength conversion capability, so that a lightpath can be established between any node pair if resources are available along the path. Each OXC is connected to an edge device like an IP router, which can be a source, or a destination of packet traffic and which can provide routing for multi hop traffic passing by that node. Our network model considers network with an initial traffic matrix and reconfiguration decisions are based on traffic changes whenever such changes are necessary.

3.3. Traffic Model
The traffic models used for simulation of the VTR algorithm for IP over WDM network are:

i. independent and identically distributed (i.i.d.) traffic model
ii. traffic cluster model

The independent and identically distributed (i.i.d) traffic model is taken to arrive QoS parameters. The i.i.d traffic model assumes uniform distribution between 0 and a maximum traffic density.

3.4. Notations
The following are the notations used in the problem formulation and in the algorithm.

\begin{itemize}
  \item $i,j$ : end nodes
  \item $s,d$ : source-destination pair
\end{itemize}

3.5. Parameters
Listed below are the parameters used in the problem.

\begin{itemize}
  \item Number of nodes in the network = $N$
  \item Number of wavelengths per fiber = $w$
  \item Capacity of each wavelength channel = $C \text{ bps}$
  \item Number of transceivers per node = $R$
  \item Average Weighted Hop count = $AWHT$
\end{itemize}

3.6. QoS Parameters
The QoS parameters considered in this research work are blocking probability message delay and throughput. These parameters are defined as follows.

i. **Blocking Probability (B):**
Blocking probability of a network at a particular instant of time is defined as the ratio of number of traffic calls blocked to the total number of traffic requests.

ii. **Message Delay (D):**
Message delay in a network at a particular instant of time is defined as the average delay incurred for a message to travel from a source node to destination node.

iii. **Throughput ($\tau$):**
Throughput of a network is defined as the ratio of number of packets received at the destination node to the number of packets transmitted from the source node.

The assumptions made for the calculation of blocking probability are as follows.

\begin{itemize}
  \item The packet arrivals are uniform and Poisson distributed at a rate of $\lambda$
\end{itemize}
ii. The packet inter-arrival times are exponentially distributed

iii. Blocked calls are cleared

The Erlang loss formula \([14]\) is given by,

\[
B = B_n + \frac{A^C}{C!} \left( 1 + \frac{A}{2!} + \frac{A^2}{3!} \right) \quad (3)
\]

which gives the blocking probability \((B)\) for the IP over WDM network, where \(A\) is the traffic load \((\lambda/\mu)\) in erlangs, \(B_n\) is the blocking probability due to reconfiguration and \(C\) is the channel capacity of one wavelength.

The throughput \((\tau)\) of the network is estimated by the formula given by,

\[
\text{Throughput} = \frac{\text{No. of packets received}}{\text{No. of packets sent}} \quad (4)
\]

The latency \((D)\) of the network is delay incurred by the data packet from a source node to a destination node and is given by,

\[
\text{Latency} = D_q + D_p + D_t + D_r \quad (5)
\]

where,

\[
D_q: \text{Queuing delay} = \frac{\sum \sum T_{ij} \beta_{ij}}{\left( C - \sum T_{ij} \right)} \quad (6)
\]

where, \(T_{ij}\) is the traffic demand between a source destination pair using the link \(ij\).

\[
D_p: \text{Propagation delay} = \frac{\text{packet length} l}{\text{channel bit rate} r} \quad (8)
\]

\[
D_t: \text{Transmission delay} = \frac{1}{(9R - (T_{ij} + X_i))} \quad (9)
\]

where, \(R\)- processing capability of router in Mbps \(T_{ij}\) is the traffic demand between a sd pair \(X_i\) is the sum of traffic routed by router \(i\) except the traffic originating at the node \(i\), and is given by,

\[
X_i = \left[ \sum T_{ij} + \sum T_{ji} + \sum T_{ij} \beta_{ij} \right] \cdot T_{ij} \quad (10)
\]

where \(\beta\) - binary variable for lightpath existence

4. HEURISTIC ALGORITHMS

In this section, a heuristic algorithm for blocking parameter driven Virtual Topology Reconfiguration (VTR) considering QoS parameters is presented.

4.1 VTR Algorithm with QoS

Input: Physical Topology; Current Virtual Topology, Traffic Demands,

Output: Reconfigured Virtual Topology

Algorithm:

for all sd pairs

compute : \(\text{WHT} = T_{sd} \cdot H_{sd}\)

end for

\[\sum \text{Pairs in non-increasing WHTs}\]

for all sd pairs

compute shortest paths using allpair shortest path algorithm

if no lightpath exists

if free wavelength available

establish lightpaths

else lightpaths deletion:

find different set of lightpaths to be deleted

sort lightpaths in non-decreasing order of load

delete the first lightpath in the set

establish lightpaths

if the topology is connected then break;

else continue

compute \(B\) for the new topology
if \( B_{new} < B_{th} \)

\( \begin{align*} 
\text{if } (N_{ch} < N_{th}) \text{ include the} \\
\text{new topology in to VT set} \\
\text{else discard the new topology}
\end{align*} \)

end for

select the VT with min \( B \)

5. RESULTS AND DISCUSSION

The heuristic algorithm for VTR is implemented for an IP/WDM network having 14 nodes with NSFNET topology [15][16] shown in fig 1. The data structures required for simulating this algorithm were written using Java [17]. The performance of the algorithm is measured for the dynamic traffic with the parameters shown in fig 2.

![Figure 1: NSF Network with 14 nodes](image)

**Parameters:**

i. Number of nodes, \( N=14 \)

ii. Number of wavelengths per link, \( w=10 \)

iii. Capacity of each wavelength = 1 unit

iv. Packet size = 512 bits

v. Channel bit rate = 1Gbps

vi. Propagation delay = 1 \( \mu \)S

vii. Distance between nodes = 1 unit

![Figure 2: NSF Network Parameters Used for Simulation](image)

The Average Weighted Hop Count measured by varying \% of change in traffic for dynamic network with i.i.d traffic model is plotted in the fig. 3. From this graph, it is observed that the AWHT is minimal compared to the two-stage reconfiguration approach given in [10]. The reduction in AWHT is due to the optimal path found by the RWA algorithm, which is described in detail in [18].

The blocking probability measured for different virtual load of the dynamic network for i.i.d traffic model is plotted in the fig. 4. From this graph, it is observed that the blocking probability is much less than that of two-stage approach.

The network throughput measured for different virtual load of the dynamic network for i.i.d traffic model is plotted in the fig. 5. From this graph, it is observed that the network throughput is maximal for the new heuristic approach till the virtual link load is 50. After that a slight decrease in throughput and reaches a steady state value.
at the load value of 100. But for the same case, the existing approach has network throughput, which is much less than that for new approach. Thus the throughput of the proposed VTR heuristic for the i.i.d. traffic model is significantly better than that of the existing approach.

The network latency measured by varying % of change in traffic for dynamic network with i.i.d traffic model is plotted in the fig. 6.

From the above graphs, it is observed that the AWHT is minimal, blocking probability is minimal, network throughput is maximal and network latency is minimal, compared to the two-stage approach given in [10].

6. CONCLUSION
In this paper, it is proposed an algorithm for Blocking Parameter driven VTR algorithm with considering QoS parameters for IP over WDM optical networks with i.i.d. traffic model. The proposed heuristic approach was validated using simulation. The simulation results show that the new approach achieves better QoS in terms of blocking probability; throughput and latency for i.i.d. traffic model for dynamic IP over WDM networks, compared to the existing two stage approach [10]. The future work includes devising VTR algorithms for distributed GMPLS networks with TE/QoS and implementing the algorithms for the above stated GMPLS networks using GLASS [19] [20].

REFERENCES


Author's Biography

Ramasamy Mariappan had received his Under Graduate Degree in Electronics and Communication Engineering from National Engineering College, Madurai Kamaraj University, INDIA in 1992. Then he has received his Post Graduate Degree in Electronics and Communication Engineering from the College of Engineering, Guindy, Anna University, Chennai, INDIA in 2000. He has more than 15 years of experience in teaching and research. Currently, he is serving as a Professor in Aarupadai Veedu Institute of Technology, Vinayaka Missions University in Tamilnadu, INDIA and he is pursuing his Research Programme. His research interests include Computer Networks, Optical Networks, Network Security, and Image Processing and guided 6 Project works for Post Graduate programme in Computer Science and
Engineering in these research areas. He has published 25 research papers in National and International conferences and three in the Journal in his research field. Further, he has participated in many National and International Conferences, Short Term Training Programmes, Faculty Development Programmes, etc. He is a life member of Computer Society of India (CSI) and Indian Society for Technical Education (ISTE). He is also a Fellow of the United Writers Association of India (FUWAI).

Dr. R. Shanmugalakshmi has published more than 40 International/ National journals. Her research area includes Image Processing, Neural Networks, Information Security and Cryptography. She has received Vijya Ratna Award from India International Friendship Society in the year of 1996, she has received Mahila Jyothi Award from Integrated Council for Socio-Economic Progress in the year of 2001 and she has received Eminent Educationalist Award from International Institute of Management, New Delhi in the year of 2008. She is member of Computer Society of India, ISTE and FIE.
Performance Computation Of Object Oriented Programs Through UML

Vipin Saxena1       Deepa Raj2

ABSTRACT
Nowadays, different kinds of processors are appearing in the business market, therefore it is necessary to judge the performance of the processors over the different kinds of programming languages. In this context, the present paper deals with the evaluation of the performance of Pentium IV architecture which is widely accepted in the global business market & performance is computed for object oriented programming languages. The main objective of this paper is to select the best object oriented programming language for long computations available in the software coding. A well known popular approach of modeling i.e. Unified Modeling Language (UML) is used for the design of Pentium IV processor architecture. UML Class, Sequence and Activity diagrams are designed and comparison is represented among the different object Oriented programming languages namely C#, Java and Visual C++.

Keywords: Pentium IV, UML, C#, VC++, Java, Performance Evaluation

1. RELATED WORK
UML [1] is a powerful modeling language used to represent the research problems visually. A lot of literature is available on modeling problems by the use of UML, but limited research papers are reported in literature on applications of UML in the field of computer architecture problems. By the use of UML, software and hardware architecture problems can be easily solved and performance can be judged after modeling of that problem. Real time system based on UML is described by Selic and Rumbaugh [2]. The first represented of UML in the field of telecommunication sector is described by Holz [3]. Drozdowski [4] explained a technique to find out the execution time for distributed application. In [5], tools and techniques for performance measurement of large distributed multi agent system are explained. Architecture of Pentium IV is reported by Alenn Hinton [6]. The computer architecture models which can be used for the further research work are available in [7]. The UML application is also done on web based application. One of the important papers on this is [8]. UML based Vehicle control system is also reported in the literature by Walther et al [9]. OMG is an important active group for inventing the different versions of the UML. The research papers on these are [10, 11] in which group describes the UML diagram based on XML Meta data specification. Performance modeling and prediction tools for parallel and distributed programs are described by Planna et al. [12, 13] and these papers also describe customizing the UML for modeling performance oriented applications. Recently Saxena et al. [14, 15] proposed the UML model with performance evaluation for the multiplex system for the process which are executing in the distributed environment and UML model with performance evaluation of the Instruction Pipeline model, respectively.

2. BACKGROUND
Let us first explain the process which may be the group or block of instructions of program, macro, sub programs and subroutines. For defining the process, there is a need.
of the processing element. The processing element is defined as a stereotype and generally used to handle the concurrent process executing in the parallel and distributed environments. The famous approach to handle the concurrent processes is Torus Topology [7]. The following figure 1 shows the definition of processing unit. The Class Diagram of process is represented in figure 2.

In _, process shows the name of multiple objects. By the use of above definition of process, in this paper, blocks of instructions are considered as a process and three types of object oriented language namely VC++ C# and JAVA are selected for judging the performance of these object oriented software programs for the blocks of instructions on the Pentium IV processor architecture, which is widely accepted processor architecture by software Industries.

C# is an Object Oriented programming language developed by Microsoft as part of .NET initiative. It is applicable for writing application in hosted and embedded system. VC++ is also an integrated development environment developed by Microsoft. It uses Microsoft Foundation Class Library (MFC) standard, which wrap the Windows API in C++ classes and provides a default application framework. On the other hand, Java is an object oriented language and many of the software designs are coded with the help of the Java language. In the present paper the performance of these three programming languages is observed on the Pentium IV architecture system by proposing a model through the UML. The main aim of this paper is to select the best object oriented programming language for writing the software codes for long computations purpose which also saves the execution time. The complete UML Diagram is designed for execution of instructions of a program. UML class diagram, UML sequence diagram and UML activity diagram are also given in the paper.

3. UML Modeling of Pentium IV
Let us consider the Pentium IV processor architecture as shown in figure 5 which is easily available in the market and widely used by the software Industries. In this architecture, the following are the four major sections:

a) In Order Front End
It is a part of machine that fetches the instruction that is to be executed next in the program and prepare them to be used later in the machine pipeline.
b) Execution Engine
In this part instructions are prepared for execution. It has several buffers which are used for smooth and reorder the flow of instruction and optimize performance as they go down to pipeline.

c) Integer and Floating Point Execution Unit
In this part instructions are actually executed. It has register file which stores the integer and floating point data operand value, and L1 data cache stores most load and store operation.

d) Memory Subsystem
It includes L2 Cache and system Bus, L2 Cache stores both Instruction and data that cannot fit in the execution trace cache and L1 data Cache. Trace Cache is primary or L1 Instruction cache of Pentium IV processor it delivers µops to the Out of Order Execution Logic. Most instructions in a program are fetched and executed from the trace cache. When there is trace cache miss then the net burst Micro architecture fetch and decode instruction from level 2 (L2) Cache. The executed and trace cache takes the already decoded µops. When the complex instruction encountered the trace cache jump into the microcode ROM which then issues the µops needed to complete the operation. After completing the work Front end of machine resumes fetching µops from the trace cache. Then instruction executes in double clocked ALU and produces required result.

A. UML Class Model
For the above Pentium IV architecture, UML class model is designed and shown in figure 6 which has following major classes- namely Process, Cache, L2_cache, L1_cache, Tc_fetch, Trace_cache, Queue, Memory_Q, Float_Q, ALU, Scheduler, Resource_alloc, Microcode_ROM, Int_Register_file, Float_Register_file, Check_branch, Instruction_Decoder, Int_schedular, Float_schedular. In this class diagram initially process is cached by the cache class which is inherited into the L1_cache and L2_cache. Instruction_Decoder is used to decode the instructions.
one by one and Trace_catch is used to load into Trace_catch. Tc_Fetch is used for fetching instruction from Trace_cache and also check the branching in the set of instructions, if instruction is complex Microcode_ROM is used to execute the instruction then load into Queue after allocating the resources, if instruction is simple then it goes into the Queue according to their type after allocating resources. Scheduler is used to schedule the instruction according to their type and as per availability of processes Arithmetic Logic Unit (ALU) class is designed to execute the instructions, then result is store in L1_cache. Same procedure will be repeated for the next set of instructions.

B. UML Sequence Diagram
For the Pentium IV architecture, sequences of instructions to be executed are arranged by means of a sequence diagram which shows the dynamic behavior of the modeled Pentium IV architecture and shown below in figure 7. In this diagram one can see that how message passing takes place among the different objects like Process, Cache, Instruction_Decoder, Tc_fetch, Resource_Alloc, Queue, Scheduler, Register_file, ALU. From the diagram one can compute the time that is used for execution of block of instructions.
C. UML Activity Diagram

For the execution of processes through Pentium IV architecture, an activity diagram for process execution is designed and shown below in figure 7. This diagram shows the steps involved in executing a program under Pentium IV.

![UML Activity Diagram Process Execution for Pentium IV](image)

4. NUMERICAL EXPERIMENTS

To judge the performance of object oriented programming languages like C#, VC++ and Java on the Pentium IV processor architecture, it is necessary to check the performance of the designed UML model for Pentium IV processor architecture. Let us consider a sequence of instructions varying from $10^1$, $10^2$, $10^3$, $10^4$ & $10^5$ lines of codes are to be executed in the five run and results are reported by taking the average technique. The following table 1 gives the total execution time of three object oriented programming languages and time is recorded in milliseconds. C#, VC++ and Java oriented software languages for executing a program of different sizes under Pentium IV processor are specially considered since the most of the software companies are developing the applications by writing software codes in these languages.

From the table 1 it is found that for the long computations on Pentium IV, Visual C++ execution time is lesser in comparison of the other two object oriented programming languages namely C# and Java. Therefore, Visual C++ is recommended on Pentium IV for long computations. These results which are recorded in the table 1 are also depicted graphically and shown in the figure 9 and 10 for $10$, $10^2$, $10^3$ and $10^4$, $10^5$, respectively.

<table>
<thead>
<tr>
<th>Lines of Code</th>
<th>C#</th>
<th>VC++</th>
<th>JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^1$</td>
<td>15</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>$10^2$</td>
<td>31</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>$10^3$</td>
<td>203</td>
<td>155</td>
<td>157</td>
</tr>
<tr>
<td>$10^4$</td>
<td>1178</td>
<td>790</td>
<td>1093</td>
</tr>
<tr>
<td>$10^5$</td>
<td>11875</td>
<td>10937</td>
<td>11762</td>
</tr>
</tbody>
</table>
5. CONCLUSION

From the above work it is concluded that UML modeling which is widely used by the scientists/researchers is excellent modeling language used to represent the scientific problem visually. The produced UML design for the Pentium IV architecture is an efficient and different kinds of object oriented programming languages are tested on this architecture and found that for long computations Visual C++ programming language is recommended. The software industries can design the code on Pentium IV architecture by the use of Visual C++. The present work further can be extended in the direction of judging the performance of other processor architectures; day by day coming into the business market around the globe; and can be tested on the multiple object oriented languages.

REFERENCES


Author’s Biography

Dr. Vipin Saxena got his M.Phil. Degree in Computer Application in 1991 & Ph.D. Degree work on Scientific Computing from University of Roorkee (renamed as Indian Institute of Technology, India) in 1997. He has more than 12 years teaching experience and 16 years research experience in the field of Scientific Computing & Software Engineering. Currently he is proposing software designs by the use of Unified Modeling Language for the various research problems related to the Software Domains & Advanced Computer Architecture. He has published more than 60 International and National publications.

Deepa Raj got her M.Sc. Degree in Computer Science from J.K. Institute of Applied Physics & Technology, Allahabad Central University, Allahabad. She has more than eight years teaching experience in field of Computer science. She is solving software designs research problems by the use of Unified Modeling Language and has several outstanding research papers in this field.
Software Maintainability Prediction Model for Object-Oriented Software Systems Based On Sensitivity - Based Linear Learning Method

Sunday Olusanya Olatunji

A B S T R A C T
This paper presented a new maintainability prediction model for an object-oriented (OO) software system based on the recently introduced learning algorithm called Sensitivity Based Linear Learning Method (SBLLM) for two-layer feedforward neural networks. As the number of object-oriented software systems increases, it becomes more important for organizations to maintain those systems effectively. However, currently only a small number of maintainability prediction models are available for object oriented systems. In this work, we develop Sensitivity Based Linear Learning maintainability prediction model for an object-oriented software system. The model was constructed using popular object-oriented metric datasets, collected from different object-oriented systems. Prediction accuracy of the model was evaluated and compared with commonly used regression-based models and also with Bayesian network based model which was earlier developed using the same datasets. Empirical results from simulation show that our SBLLM based model produced promising results in term of prediction accuracy measures authorized in OO software maintainability literatures, better than most of the other earlier implemented models on the same datasets.


1. INTRODUCTION
Software maintainability is the process of modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment. Maintaining and enhancing the reliability of software during maintenance requires that software engineers understand how various components of a design interact. People usually think of software maintenance as beginning when the product is delivered to the client. While this is formally true, in fact decisions that affect the maintenance of the product are made from the earliest stage of design.

Software maintenance is classified into four types: corrective, adaptive, perfective and preventive. Corrective maintenance refers to fixing a program. Adaptive maintenance refers to modifications that adapt to changes in the data environment, such as new product codes or new file organization or changes in the hardware of software environments. Perfective maintenance refers to enhancements: making the product better, faster, smaller, better documented, cleaner structured, with more functions or reports. The preventive maintenance is defined as the work that is done in order to try to prevent malfunctions or improve maintainability.
When a software system is not designed for maintenance, it exhibits a lack of stability under change. A modification in one part of the system has side effects that ripple throughout the system. Thus, the main challenges in software maintenance are to understand existing software and to make changes without introducing new bugs.

It is arguable that many object-oriented (OO) software systems are currently in use. It is also arguable that the growing popularity of OO programming languages, such as Java, as well as the increasing number of software development tools supporting the Unified Modelling Language (UML), encourages more OO systems to be developed at present and in the future. Hence it is important that those systems are maintained effectively and efficiently. A software maintainability prediction model enables organizations to predict maintainability of a software system and assists them with managing maintenance resources.

In addition, if an accurate maintainability prediction model is available for a software system, a defensive design can be adopted. This would minimize, or at least reduce future maintenance effort of the system. Maintainability of a software system can be measured in different ways. Maintainability could be measured as the number of changes made to the code during a maintenance period or be measured as effort to make those changes. The predictive model is called a maintenance effort prediction model if maintainability is measured as effort. Unfortunately, the number of software maintainability prediction models including maintenance effort prediction models, is currently very small in the literature.

In this research work, we developed a new maintainability prediction model for an object-oriented software system based on the recently introduced learning algorithm called Sensitivity Based Linear Learning Method (SBLLM). It is a learning technique for two-layer feed forward neural networks based on sensitivity analysis, which uses a linear training algorithm for each of the two layers. In theory, this algorithm tends to provide good generalization performance at extremely fast learning speed. The experimental results, found in literatures, based on a few artificial and real benchmark function approximation and classification problems including very large complex applications, and particularly the empirical results from this study, demonstrated that the SBLLM can produce good generalization performance in most cases and can learn thousands of times faster than conventional popular learning algorithms for feed-forward neural networks.

Despite the importance of software maintenance, little work has been done as regards developing predictive models for software maintainability, particularly object-oriented software system, which is evident in the fewer number of software maintainability prediction models, that are currently found in the literature.

In view of this, we have developed a new maintainability prediction model for an object-oriented software system based on the recently introduced learning algorithm called Sensitivity Based Linear Learning Method. Implementation was carried out on representative datasets related to the target systems. Furthermore, we performed comparative analysis between our model and the models presented in, Koten (2006), which include Regression-Based and Bayesian Network Based models, in terms of their performance measures values, as recommended in the literatures.

Furthermore, the usefulness of the SBLLMs in the area of software engineering and, in particular, maintainability prediction for an object-oriented software system, has been made clearer by describing both the steps and the use of SBLLM as an artificial intelligence modeling
approach for predicting the maintainability of object-oriented software system.

The rest of this paper is organized as follows. Section 2 contains review of related earlier works. Section 3 discusses Sample predictive modelling techniques, and also describes the main modelling technique used: SBLLM. Section 4 presents the OO software data sets and the metrics used in our study and their descriptions. Section 5 model evaluation that include model validation approach and prediction accuracy measures used. Section 6 contains empirical results, comparison with other models and discussions. Section 7 concludes the paper.

2. RELATED WORK

Several object oriented software maintainability prediction models were developed of recent; and they are mostly characterized by low prediction accuracies Lucia et al. (2005). Regression techniques have been thoroughly utilized by Li and Henry (1993), Fioravanti and Nesi (2001), and Misra(2005) to predict maintainability of object oriented software systems. Some recent work have been done using artificial neural networks and some other artificial intelligence techniques such as Bayesian Belief Networks (BBN), van and Gray (2006), and Multivariate adaptive regression splines (MARS), Zhou and Leung (2007).

Variants of artificial neural networks were also employed in predicting the maintainability effort of object oriented software systems. Feed forward neural network and General Regression neural network (GRNN) were used by Quah et al. (2003) to predict the maintainability effort for object oriented software systems using object oriented metrics. On the other hand, back-propagation multilayer perceptron (BP-MLP) have been used by Mahaweerawat, Sophatsathit, Lursinsap Musilek (2003) to predict faulty classes in object oriented software. In the same research work, they used radial basis function networks (RBF) to predict the type of fault a faulty class has.

Bayesian Belief Networks (BBN) was first suggested as a novel approach for software quality prediction by Fenton et al. (2002) and Fenton and Neil (1999, 2000). They build their conjecture based on Bayesian Belief Networks’ ability in handling uncertainties, incorporating expert knowledge, and modeling the complex relationships among variables. However, a number of researchers, have pointed out several limitations of Bayesian Belief Networks when they are applied as a model for object oriented software quality and maintainability prediction Ma et al. (2006), Weaver (2003), and Yu et al. (2002). Recently, a special type of Bayesian Belief Networks called Naïve-Bayes classifier was used by van and Gray (2006) to implement a Bayesian-Belief-Networks-based software maintainability prediction model. Although their results showed that their model give better results than regression-based techniques for some datasets, the model is still inferior to regression-based techniques for some other datasets.

3. SAMPLE MODELING TECHNIQUES

3.1 Regression Based Models

Regression models are used to predict one variable from one or more other variables. Regression models provide the scientist with a powerful tool, allowing predictions about past, present, or future events to be made with information about past or present events.

3.1.1 Multiple Linear Regression Model

Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent variable y.
The regression line for \( p \) explanatory variables is defined to be
\[
\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p.
\]
This line describes how the mean response \( \mu_y \) changes with the explanatory variables. The observed values for \( y \) vary about their means and are assumed to have the same standard deviation. Formally, the model for multiple linear regression given \( n \) observations is
\[
y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i.
\]
for \( i = 1, 2, \ldots, n \)

Where \( \epsilon_i \) is notation for model deviation.

One approach to simplifying multiple regression equations is the stepwise procedures. These include forward selection, backwards elimination, and stepwise regression. They add or remove variables one-at-a-time until some stopping rule is satisfied.

**Forward Selection:** Forward selection starts with an empty model. The variable that has the smallest P value when it is the only predictor in the regression equation is placed in the model. Each subsequent step adds the variable that has the smallest P value in the presence of the predictors already in the equation. Variables are added one-at-a-time as long as their P values are small enough, typically less than 0.05 or 0.10.

**Backward Elimination:** It starts with all of the predictors in the model. The variable that is least significant that is, the one with the largest P value is removed and the model is refitted. Each subsequent step removes the least significant variable in the model until all remaining variables have individual P values smaller than some value, such as 0.05 or 0.10.

**Stepwise regression** This approach is similar to forward selection except that variables are removed from the model if they become non significant as other predictors are added.

**Backwards Elimination:** has an advantage over forward selection and stepwise regression because it is possible for a set of variables to have considerable predictive capability rather than any individual subset. Forward selection and stepwise regression will fail to identify them because sometimes variables don’t predict well individually and Backward elimination starts with everything in the model, so their joint predictive capability will be seen.

### 3.2 Bayesian Networks

A Bayesian network consists of nodes interconnected by the directed links forming directed acyclic graph. In this graph, nodes represent random variables (RVs) and links correspond to direct probabilistic influences. The RVs correspond to important attributes of the modeled system which exemplifying the system’s behavior. Directed connection between the two nodes indicates a casual effect.

The structure of directed acyclic graph states that each node is independent of all its non descendants conditioned on its parent nodes. In other words, the Bayesian Network represents the conditional probability distribution \( P(Y/X_1, \ldots, X_n) \) which is used to quantify the strength of variables \( X_i \) on the variable \( Y \), Nodes \( X_i \) are called the parents of \( Y \) and \( Y \) is called a child of each \( X_i \). This should be noted that outcomes of the events for the variables \( X_i \) have an influence on the outcome of the event \( Y \).

### 3.3 Sensitivity Based Linear Learning Method (SBLLM)

Castillo et al (2006), proposed a new learning scheme in order to both speed up and avoid local minima convergence of the existing backpropagation learning...
technique. This new learning strategy is called the Sensitivity Based Linear Learning (SBLLM) scheme. It is a learning technique for two-layer feedforward neural networks based on sensitivity analysis, which uses a linear training algorithm for each of the two layers. First, random values are assigned to the outputs of the first layer; later, these initial values are updated based on sensitivity formulas, which use the weights in each of the layers; the process is repeated until convergence. Since these weights are learnt solving a linear system of equations, there is an important saving in computational time. The method also gives the local sensitivities of the least square errors with respect to input and output data, with no extra computational cost, because the necessary information becomes available without extra calculations. This new scheme can also be used to provide an initial set of weights, which significantly improves the behavior of other learning algorithms. The full theoretical basis for SBLLM and its performance has been demonstrated in Castillo et al (2006), which contained its application to several learning problems examples in which it is compared with several learning algorithms and well known data sets. The results have shown a learning speed generally faster than other existing methods. In addition, it can be used as an initialization tool for other well known methods with significant improvements.

Sensitivity analysis is a very useful technique for deriving how and how much the solution to a given problem depends on data, see Castillo et al., 1997, 1999, 2000 and the references therein for more details. However, in Castillo et al (2006) it was shown that sensitivity formulas can also be used as a novel supervised learning algorithm for two-layer feedforward neural networks that presents a high convergence speed. Generally, SBLLM process is based on the use of the sensitivities of each layer’s parameters with respect to its inputs and outputs and also on the use of independent systems of linear equations for each layer to obtain the optimal values of its parameters. In addition, it gives the sensitivities of the sum of squared errors with respect to the input and output data.

3.3.1 How Sensitivity Based Linear Learning Method (SBLLM) Works

Consider the two-layer feedforward neural network in Figure 1, where I is the number of inputs, J the number of outputs, K the number of hidden units, x0s =1, z0s =1, S the number of data samples and the superscripts (1) and (2) are used to refer to the first and second layer, respectively. This network can be considered to be composed of two one-layer neural networks as it is shown in Figure 2. For this one layer neural network, to learn the weights wji, one can minimize the sum of squared errors before the nonlinear activation functions, Castillo et al. (2002), that is,

\[ Q = \sum_{s=1}^{S} \sum_{p=1}^{P} \sum_{j=1}^{J} (y_{j} - f_{j}(x_{p}))^2. \]  

(1)

this leads to the system of equations:

\[ \sum_{p} A_{pj} w_{ji} = b_{j}, \quad p = 0, 1, \ldots, I; \quad \forall j, \]  

(2)

where
Therefore, assuming that the intermediate layer outputs $z$ are known, using equation (1), a new cost function for the two-layer feedforward neural network in Figure 1 is defined as:

$$Q(z) = Q^1(z) + Q^2(z) = \sum_{s=0}^{S} \left[ \sum_{k=1}^{K} \sum_{i=0}^{I} w^{(1)}_{ik} x_i - f^{(1)}_{s}(z_{sk}) \right]$$

Thus, using the outputs $z_k$’s we can learn, for each layer independently, the weights $w^{(1)}_{ik}$ and $w^{(2)}_{jk}$ by solving the corresponding linear system of equations (2). For the neural network shown in Figure 1, according to Castillo et al. (2001, 2004, and 2006), the sensitivities of the new cost function, $Q$, with respect to the output and input data can be obtained as:

$$\frac{\partial Q}{\partial y_{pq}} = \sum_{j=0}^{J} \sum_{q=0}^{Q} \left[ w^{(2)}_{pq} f^{(1)}_{j}(y_{jq}) \right] w^{(2)}_{pq}$$

Thus, the sensitivities with respect to $z_k$, for the two-layer feedforward neural network in Figure 1 are calculated as:

$$\frac{\partial Q}{\partial z_{ks}} = \frac{\partial Q^{(1)}}{\partial z_{ks}} + \frac{\partial Q^{(2)}}{\partial z_{ks}} =$$

$$2 \left( \sum_{i=0}^{I} w^{(1)}_{ij} x_i - f^{(1)}_{s}(z_{sk}) \right)$$

$$+ 2 \sum_{j=0}^{J} \left( \sum_{r=0}^{R} w^{(2)}_{jr} z_r - f^{(2)}_{s}(y_{jq}) \right) w^{(2)}_{jr}$$

with $k = 1, \ldots, K$, as $z_k = 1$’s. After this, the values of the intermediate outputs $z$ are modified using the Taylor series approximation:

$$Q(z + \Delta z) = Q(z) + \sum_{s=0}^{S} \sum_{k=1}^{K} \frac{\partial Q(z)}{\partial z_{sk}} \Delta z_{sk} \approx 0,$$

which leads to the following increments:

$$\Delta z = -\rho \frac{Q(z)}{\|\nabla Q\|^2} \nabla Q,$$

where $\rho$ is a relaxation factor or step size. The Sensitivity-Based Linear Learning scheme is summarized in the following algorithmic steps.

### 3.3.2 SBLLM Learning Process

The training algorithm of the SBLLM technique can be summarized in the following algorithmic steps:

1. **Input** - The inputs to the system, which is the available or simulated data (training) set (input, $x_i$, and desired data, $y_j$), two threshold errors ($\varepsilon$ and $'\varepsilon$) to control both convergence and a step size $\rho$.
Output: The output results of the SBLLM system are the weights of the two layers and the sensitivities of the sum of squared errors with respect to input and output data.

Step 0: Initialization. Assign to the outputs of the intermediate layer the output associated with some random weights $w(1)(0)$ plus a small random error, that is,

$$w_i^{(1)}(0) = w_i^{(1)}(0) + \eta, \quad i = 1, \ldots, I$$

where $\eta$ is a small number, and initialize the sum of squared errors ($Q_{\text{previous}}$) and mean-squared errors ($\text{MSE}_{\text{previous}}$) to some large number, where MSE measures the error between the obtained and the desired output.

Step 1: Sub-problem solution. Learn the weights of layers 1 and 2 and the associated sensitivities solving the corresponding systems of equations, that is,

$$A^{(1)}_{p_i} = \sum_{s=1}^{S} x_{p_s} x_{p_s}, \quad b^{(1)}_{p_k} = \sum_{s=1}^{S} f^{(1)-1}(z_{p_s}) x_{p_s}, \quad p = 0, 1, \ldots, I; k = 1, 2, \ldots, K,$$

and

$$A^{(2)}_{q_k} = \sum_{s=1}^{S} z_{q_s} z_{q_s}, \quad b^{(2)}_{q_j} = \sum_{s=1}^{S} f^{(2)-1}(y_{q_s}) z_{q_s}, q = 0, 1, \ldots, K; \forall j.$$  

Step 2: Evaluate the sum of squared errors. Evaluate $Q$ using

$$Q(z) = Q^{(1)}(z) + Q^{(2)}(z) =$$

$$= \sum_{s=1}^{S} \left[ \sum_{k=1}^{K} \left( \sum_{i=0}^{I} W^{(1)}_{ki} x_i - f^{(1)-1}(z_{ks}) \right)^2 + \sum_{j=1}^{J} \left( \sum_{k=0}^{K} W^{(2)}_{jk} z_{ks} - f^{(2)-1}(y_{js}) \right)^2 \right],$$

and evaluate also the MSE.

Step 3: Convergence checking. If $|Q - Q_{\text{previous}}| < \varepsilon$ or $|\text{MSE}_{\text{previous}} - \text{MSE}| < \varepsilon$ stop and return the weights and the sensitivities. Otherwise, continue with Step 4.

Step 4: Check improvement of $Q$. If $Q > Q_{\text{previous}}$ reduce the value of $\rho$, that is, $\rho = \rho / 2$ and return to the previous position, that is, restore the weights, $z = z_{\text{previous}}$, $Q = Q_{\text{previous}}$ and go to Step 5. Otherwise, store the values of $Q$ and $z$, that is, $Q_{\text{previous}} = Q$, $\text{MSE}_{\text{previous}} = \text{MSE}$ and $z_{\text{previous}} = z$ and obtain the sensitivities using:

$$\sum_{s=1}^{S} f^{(1)}_{s} A_{p_i}^{(1)} \left( \sum_{j=0}^{J} W^{(1)}_{ji} x_j - f^{(1)-1}(z_{ps}) \right) \to U(-\eta, \eta);$$

and

$$\sum_{s=1}^{S} z_{q_j} W^{(2)}_{js} \left( f^{(2)-1}(y_{q_s}) \right) \to U(-\eta, \eta);$$

$$w_{jk}^{(2)} = \sum_{s=1}^{S} \left( w^{(2)}_{jk} - f^{(2)}_{jk}(y_{js}) \right), \quad k = 1, \ldots, K.$$  

Step 5: Update intermediate outputs. Using the Taylor series approximation in equation (Eq.s3), update the intermediate outputs as

$$z = z - \rho \frac{\partial Q(z)}{\partial Q} \nabla Q,$$

and go to Step 1.
3.4.3 Advantages of SBLLM

SBLLM offers an interesting combination of speed, reliability and simplicity. In addition, based on the results obtained from the real-world experiments using the SBLLM learning algorithm, there are four main advantages of the SBLLM that can be summarized as follows, Castillo et al. (2006):

1. High speed in reaching the minimum error: It was demonstrated in Castillo et al 2006, that in all cases the SBLLM obtains its minimum MSE (MSE$_{min}$) just before the first four iterations and also sooner than the rest of the algorithms examined together. SBLLM gets its minimum error in an epoch for which the other algorithms are far from similar MSE values.

2. Good performance: It can be deduced that not only that SBLLM stabilizes soon, but also the minimum MSE that it reaches is quite good and comparable to that obtained by the second order methods. Other methods compared to it never succeeded in attaining this minimum MSE (MSE$_{min}$) before the maximum number of epochs.

3. Homogeneous behavior: The SBLLM learning curve stabilizes soon as it is demonstrated in Castillo et al. (2006). The SBLLM behaves homogeneously not only if we consider just the end of the learning process, but also during the whole process, in such a way that very similar learning curves where obtained for all iterations of different experiments.

4. SBLLM as Initialization Method: The other good aspect of SBLLM is that it has been used as initialization method in conjunction with other learning algorithm often with better and accurate results. In this case it always achieves a faster convergence speed, obtains a very good initial point, and thus a very low MSE in a few epochs of training. In addition, the SBLLM helps the learning algorithms to obtain a more homogeneous Mean square error (MSE) at the end of the training process. Therefore, when SBLLM is used as an initialization method, it significantly improves the performance of a learning algorithm.

4. DATA SETS

In this work, we made use of OO software datasets published by Li and Henry (1993). In this section, we describe the data set used for this study. We first introduce the metrics under study and then give some statistical analysis of the metrics that were investigated.

4.1. Studied Metrics

This study makes use of two OO software data sets published by Li and Henry (1993). These metric data were collected from a total of 110 classes in two OO software systems The first data set, UIMS, contains the metric data of 39 classes collected from a user interface management system (UIMS). The second data set, QUES, contains the metric data of 71 classes collected from a quality evaluation system (QUES). Both systems were implemented in Ada.

The datasets consist of five C&K metrics: DIT, NOC, RFC, LCOM and WMC, and four L&H metrics: MPC, DAC, NOM and SIZE2, as well as SIZE1, which is a traditional lines of code size metric. Maintainability was measured in CHANGE metric by counting the number of lines in the code, which were changed during a three- year maintenance period. Neither UIMS nor QUES datasets contain actual maintenance effort data. The description of each metric is given in the table below:
Table 1: Description Of Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMC (Weighted methods per class)</td>
<td>The sum of McCabe’s cyclomatic complexity of all local methods in a given class</td>
</tr>
<tr>
<td>DIT (Depth of inheritance tree)</td>
<td>The length of the longest path from a given class to the root in the inheritance hierarchy</td>
</tr>
<tr>
<td>RFC (Response for a class)</td>
<td>The number of methods that can potentially be executed in response to a message being received by an object of a given class</td>
</tr>
<tr>
<td>NOC (Number of children)</td>
<td>The number of classes that directly inherit from a given class, i.e., number of direct sub-classes that the class has</td>
</tr>
<tr>
<td>LCOM (Lack of cohesion in methods)</td>
<td>The number of pairs of local methods in a given class using no attribute in common, number of disjoint sets of local methods, i.e., number of sets of local methods that do not interact with each other, in the class</td>
</tr>
<tr>
<td>MPC (Message-passing coupling)</td>
<td>The number of send statements defined in a given class</td>
</tr>
<tr>
<td>DAC (Data abstraction coupling)</td>
<td>The number of abstract data types defined in a given class</td>
</tr>
<tr>
<td>NOM (Number of methods)</td>
<td>The number of methods implemented within a given class</td>
</tr>
<tr>
<td>SIZE1 (Lines of code)</td>
<td>The number of semicolons in a given class</td>
</tr>
<tr>
<td>SIZE2 (Number of properties)</td>
<td>The total number of attributes and the number of local methods in a given class</td>
</tr>
<tr>
<td>CHANGE (Number of lines changed in the class)</td>
<td>Insertion and deletion are independently counted as 1, change of the contents is counted as 2</td>
</tr>
</tbody>
</table>

DIT, NOC, RFC, LCOM, WMC, MPC, DAC, NOM, SIZE2, and SIZE1 are the features that are combined and made used of to predict the attribute change. QUES data set has 71 sample cases, whereas UIMS has 39 sample cases.
### 4.2 Characteristics of the Datasets

**Table 2: Descriptive Statistics Of The UIMS Data Set**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Maximum</th>
<th>75%</th>
<th>Median</th>
<th>25%</th>
<th>Minimum</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Skewness</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMC</td>
<td>69</td>
<td>12</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>11.38</td>
<td>15.90</td>
<td>2.03</td>
</tr>
<tr>
<td>DIT</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2.15</td>
<td>0.90</td>
<td>-0.54</td>
</tr>
<tr>
<td>RFC</td>
<td>101</td>
<td>30</td>
<td>17</td>
<td>11</td>
<td>2</td>
<td>23.21</td>
<td>20.19</td>
<td>2.00</td>
</tr>
<tr>
<td>NOC</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.95</td>
<td>2.01</td>
<td>2.24</td>
</tr>
<tr>
<td>LCOM</td>
<td>31</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>7.49</td>
<td>6.11</td>
<td>2.49</td>
</tr>
<tr>
<td>MPC</td>
<td>12</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4.33</td>
<td>3.41</td>
<td>0.731</td>
</tr>
<tr>
<td>DAC</td>
<td>21</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.41</td>
<td>4.00</td>
<td>3.33</td>
</tr>
<tr>
<td>NOM</td>
<td>40</td>
<td>13</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>11.38</td>
<td>10.21</td>
<td>1.67</td>
</tr>
<tr>
<td>SIZE1</td>
<td>439</td>
<td>131</td>
<td>74</td>
<td>27</td>
<td>4</td>
<td>106.44</td>
<td>114.65</td>
<td>1.71</td>
</tr>
<tr>
<td>SIZE2</td>
<td>61</td>
<td>16</td>
<td>9</td>
<td>6</td>
<td>1</td>
<td>13.97</td>
<td>13.47</td>
<td>1.89</td>
</tr>
<tr>
<td>CHANGE</td>
<td>289</td>
<td>39</td>
<td>18</td>
<td>10</td>
<td>2</td>
<td>46.82</td>
<td>71.89</td>
<td>2.29</td>
</tr>
</tbody>
</table>
Table 3: Descriptive Statistics of The QUES Data Set

<table>
<thead>
<tr>
<th>Metric</th>
<th>Maximum</th>
<th>75%</th>
<th>Median</th>
<th>25%</th>
<th>Minimum</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Skewness</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMC</td>
<td>83</td>
<td>22</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>14.96</td>
<td>17.06</td>
<td>1.77</td>
</tr>
<tr>
<td>DIT</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1.92</td>
<td>0.53</td>
<td>-0.10</td>
</tr>
<tr>
<td>RFC</td>
<td>156</td>
<td>62</td>
<td>40</td>
<td>34</td>
<td>17</td>
<td>54.44</td>
<td>32.62</td>
<td>1.62</td>
</tr>
<tr>
<td>NOC</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>LCOM</td>
<td>33</td>
<td>14</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>9.18</td>
<td>7.34</td>
<td>1.35</td>
</tr>
<tr>
<td>MPC</td>
<td>42</td>
<td>21</td>
<td>17</td>
<td>12</td>
<td>2</td>
<td>17.75</td>
<td>8.33</td>
<td>0.88</td>
</tr>
<tr>
<td>DAC</td>
<td>25</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3.44</td>
<td>3.91</td>
<td>2.99</td>
</tr>
<tr>
<td>NOM</td>
<td>57</td>
<td>21</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>13.41</td>
<td>12.00</td>
<td>1.39</td>
</tr>
<tr>
<td>SIZE1</td>
<td>1009</td>
<td>333</td>
<td>211</td>
<td>172</td>
<td>115</td>
<td>275.58</td>
<td>171.60</td>
<td>2.11</td>
</tr>
<tr>
<td>SIZE2</td>
<td>82</td>
<td>25</td>
<td>10</td>
<td>7</td>
<td>4</td>
<td>18.03</td>
<td>15.21</td>
<td>1.71</td>
</tr>
<tr>
<td>CHANGE</td>
<td>217</td>
<td>85</td>
<td>52</td>
<td>35</td>
<td>6</td>
<td>64.23</td>
<td>43.13</td>
<td>1.36</td>
</tr>
</tbody>
</table>

5. Model Evaluation

5.1 Model Validation Approach

The available data set, for each data set, were divided into two parts. One part was used as a training set, for constructing a maintainability prediction model. The other part was used for testing to determine the prediction ability of the developed model. Although there are many different ways to split a given dataset, we have chosen to use the stratify sampling approach in breaking the datasets due to its ability to break data randomly with a resultant balanced division based on the supplied percentage. The division, for instance could be 70% for training set and 30% for testing set. In this work, we selected 70% of the data for building the model (internal validation) and 30% of the data for testing/ validation (external validation or cross-validation criterion). We repeat both internal and external validation processes for 1000 times to have a fair partition through the entire process operations.

We also evaluate and compare our developed model with other OO software maintainability prediction models,
sited earlier, quantitatively, using the following prediction accuracy measures recommended in the literatures: absolute residual (Ab.Res.), the magnitude of relative error (MRE) and the proportion of the predicted values that have MRE less than or equal to a specified value suggested in the literatures (pred measures). Details of all these measures of performance will be provided shortly.

5.2 Prediction Accuracy Measures

In this paper, we compared the software maintainability prediction models using the following prediction accuracy measures: absolute residual (Abs Res), the magnitude of relative error (MRE) and Pred measures.

The Ab.Res. is the absolute value of residual evaluated by:

\[ Ab\text{.Res.} = \text{abs} (\text{actual value} - \text{predicted value}) \]

In this paper, the sum of the absolute residuals (Sum Ab.Res.), the median of the absolute residuals (Med.Ab.Res.) and the standard deviation of the absolute residuals (SD Ab.Res.) are used. The Sum Ab.Res. measures the total residuals over the dataset. The Med.Ab.Res. measures the central tendency of the residual distribution. The Med.Ab.Res. is chosen to be a measure of the central tendency because the residual distribution is usually skewed in software datasets. The SD Ab.Res. measures the dispersion of the residual distribution.

MRE is a normalized measure of the discrepancy between actual values and predicted values given by

\[ MRE = \text{abs} (\text{actual value} - \text{predicted value}) / \text{actual value} \]

The Max.MRE measures the maximum relative discrepancy, which is equivalent to the maximum error relative to the actual effort in the prediction. The mean of MRE, the mean magnitude of relative error (MMRE):

\[ MMRE = \frac{1}{n} \sum_{i=1}^{n} MRE_i \]

According to Fenton and Pfleeger (1997), Pred is a measure of what proportion of the predicted values have MRE less than or equal to a specified value, given by:

\[ \text{Pred} (q) = \frac{k}{n} \]

where \( q \) is the specified value, \( k \) is the number of cases whose MRE is less than or equal to \( q \) and \( n \) is the total number of cases in the dataset.

According to Conte and Dunsmore (1986), and MacDonell(1997), in order for an effort prediction model to be considered accurate, MMRE \( \leq 0.25 \) and/or either \( \text{pred}(0.25) \geq 0.75 \) or \( \text{pred}(0.30) \geq 0.70 \). These are the suggested criteria in literature as far as effort prediction is concerned.

6. Empirical Results And Comparison

Below are tables and figures showing the results of our newly developed model in comparison to the other earlier models used on the same datasets.

6.1 Results from QUES dataset

Table 3 shows the values of the prediction accuracy measures achieved by each of the maintainability prediction models for the QUES dataset. In order for an effort prediction model to be considered accurate, either MMRE \( \leq 0.25 \) and/or either \( \text{pred}(0.25) \geq 0.75 \) or \( \text{pred}(0.30) \geq 0.70 \), needed to be achieved, Conte and Dunsmore (1986), and MacDonell(1997). Hence the closer a model’s prediction accuracy measure value is to these baseline values, the better. Since Table 3 shows that the SBLLM model has achieved MMRE value of 0.348, the pred(0.25) value of 0.5 and the pred(0.30) value of
0.56. It is clear from these, that the SBLLM model is the only one that is very close to the required values for all the three essential prediction measures, hence it is the best among all the presented models. It outperform all the other model in terms of all the predictive measures used.

In comparison with the UIMS dataset, the performance on QUES dataset is far better than that on UIMS. This indicates that the performance of the SBLLM models may vary depending on the characteristics of the dataset and/or depending on what prediction accuracy measure is used.

Table 4: Prediction Accuracy For The QUES Dataset

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian network</td>
<td>1.592</td>
<td>0.452</td>
<td>0.391</td>
<td>0.430</td>
<td>686.610</td>
<td>17.560</td>
<td>31.506</td>
</tr>
<tr>
<td>Regression Tree</td>
<td>2.104</td>
<td>0.493</td>
<td>0.352</td>
<td>0.383</td>
<td>615.543</td>
<td>19.809</td>
<td>25.400</td>
</tr>
<tr>
<td>Backward Elimination</td>
<td>1.418</td>
<td>0.403</td>
<td>0.396</td>
<td>0.461</td>
<td>507.984</td>
<td>17.396</td>
<td>19.696</td>
</tr>
<tr>
<td>Stepwise Selection</td>
<td>1.471</td>
<td>0.392</td>
<td>0.422</td>
<td>0.500</td>
<td>498.675</td>
<td>16.726</td>
<td>20.267</td>
</tr>
<tr>
<td>SBLLM</td>
<td>1.713</td>
<td>0.348</td>
<td>0.5</td>
<td>0.56</td>
<td>778.437</td>
<td>11.274</td>
<td>16.258</td>
</tr>
</tbody>
</table>

6.2 Results From UIMS Dataset

Table 3 shows the values of the prediction accuracy measures achieved by each of the maintainability prediction models for the UIMS dataset. From the results presented, the SBLLM model has achieved the MMRE value of 1.966, the pred (0.25) value of 0.179 and the pred (0.30) value of 0.25. These values compete favorably among all the five models presented. Specifically in term of Sum.Abs. value, it is the best among all the models and we can see that there is strong evidence that the SBLLM model’s value is significantly lower and thus, better than those of the other models. In term of pred(0.30), it is the second best model after Bayesian network. In addition, it is also the best in term of Med. Ab.Res. and SD. Ab.Res.

Though the performance of SBLLM on UIMS dataset is low compared to its performance on QUES dataset, yet its performance compared to other models on same dataset is competitive and encouraging.
6.3 Discussion

With the exception of SBLLM that has values closer to satisfying the stated criteria, particularly on QUES dataset, none of the other prediction models presented get closer to satisfying any of the criteria of an accurate prediction model cited earlier. However, it is reported that prediction accuracy of software maintenance effort prediction models are often low and thus, it is very difficult to satisfy the criteria, Lucia et al. (2005).

Thus, we are concluding that SBLLM model presented in this paper can predict maintainability of the OO software systems reasonably well to an acceptable degree. This submission of ours is as a result of the fact that only SBLLM model has been able to consistently perform better by having values closer to satisfying the criteria laid down in literature particularly on the QUES dataset.

For UIMS datasets, whenever the SBLLM model’s prediction accuracy measure is not better than the best among the other models, it has been reasonably competitive against the best models.

Therefore, we are concluding that the prediction accuracy of the SBLLM model is better than, or at least, is competitive against the Bayesian network model and the regression based models. These outcomes have confirmed that SBLLM is indeed a useful modeling technique for software maintainability prediction, although further studies are required to realize its full potentials.

7. Conclusion

An SBLLM OO software maintainability prediction model has been constructed using the OO software metric data in Li and Henry datasets, Li and Henry (1993). The prediction accuracy of the model is evaluated and compared with the Bayesian network model, regression tree model and the multiple linear regression models using the prediction accuracy measures: the absolute residuals, MRE and pred measures. The results indicate that SBLLM model can reliably predict maintainability of the OO software systems. The SBLLM model has achieved significantly better prediction accuracy, than the other models, particularly on QUES dataset. Also, for UIMS datasets, whenever the SBLLM model’s prediction accuracy measure is not better than the best among the other models, it has been reasonably competitive against the best models.

For better visualization of results, the prediction accuracy measures for both UIMS and QUES datasets are depicted in figures 5 and 6 respectively.
The results in this paper also suggest that the prediction accuracy of the SBLLM model may vary depending on the characteristics of dataset and/or the prediction accuracy measure used. This provides an interesting direction for future studies.

Figure 5: Charts Depicting The Prediction Accuracy For The QUES Dataset
Figure 6: Charts Depicting The Prediction Accuracy For The UIMS Dataset
REFERENCES


[16] S.G. MacDonell, “Establishing relationships between specification size and software process


Software Maintainability Prediction Model for Object-Oriented Software Systems Based On Sensitivity-Based Linear Learning Method


Author’s Biography

Sunday Olusanya Olatunji received the B.Sc. (Hons) Degree in Computer Science, Ondo State University (Now University of Ado Ekiti), Nigeria in 1999. He received M.Sc. Degree in Computer Science, University Of Ibadan, Nigeria in 2003. He received another M.Sc. Degree in Information and Computer Science, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia in 2008. He is currently pursuing his Ph.D in Computer Science. He has been a Lecturer in Computer Science Department, Ondo State University, Akungba Akoko, Nigeria, since 2001, where he is presently on study leave to obtain his Ph.D. He is a member of ACM and IEEE. He has participated in numerous research projects in KFUPM including those with ARAMCO oil and gas Company.
Software For Design And Layout Of Drip And Sprinkler Irrigation Systems

Karan Singh¹ K.S. Reddy²

¹Senior Scientist, Computer Applications, Central Institute of Agricultural Engineering, Nabi Bagh, Berasia Road, Bhopal, M.P, PIN – 462 038. E-mail: ksingh@ciae.res.in
²Principal Scientist, Soil, Water & Drainage Engineering, Central Research Institute for Dryland Agriculture, Hyderabad, A.P.

ABSTRACT

A reliable and suitable irrigation water supply can result in vast improvements in agricultural production and assure the economic vitality of the region. Drip and sprinkler irrigation systems are relatively new and very efficient method for placement of water to the crop position. The design and layout of drip and sprinkler irrigation systems consist of the collection of data from different sources with respect to various parameters to arrive at a feasible and practicable design for adoption in the field. It requires tremendous efforts and energy, planning, money to pool the information for the proper design of the systems and also, it is time consuming and cumbersome. Most of the designs require both qualitative and quantitative information to arrive at possible design from the reports and literature. The effort has been made in the developed software to integrate both design procedures and databases. This saves a lot of time and various design options can be prepared and compared before laying the final design in the field. The software for design and layout of drip and sprinkler irrigation systems has been developed using Microsoft Visual Basic 6.0 as front end and Microsoft Access as back end. The user need to provide the basic information regarding drips and sprinklers such as plot size and crop to be grown, soil type, available discharge, differences in topographic height, available pumps, water quality etc to start the design process. The software allows the user to select the various parameters through different screens and finally generate a complete design layout and approximate cost of the system based on the selected parameters.

Keywords: Design And Layout, Drip And Sprinkler Irrigation Systems, Software.

1. INTRODUCTION

India has 2.4 per cent of land mass and 4 per cent fresh water resources of the world, but supports 17 per cent of the world population. Since independence, therefore, per capita land availability has dwindled from 0.48 ha to 0.15 ha and water availability has been reduced from 5300 cu m to 1500 cu m. In India, 91.6 per cent of the water used for irrigation purpose as compared to 84 per cent in Asia and 71 per cent in the world (Irrigation in Asia in figures, Water Report No. 18, FAO, 1999). The revised National Water Policy, which was formulated in 2002 has also taken cognizance of the necessity of proper irrigation water management. The expected climatic change in the 21st century will have pronounced negative impact on overall water availability and its quality. Thus, the importance of water management and the need to adopt modern techniques of irrigation to increase the yield and water use efficiencies for most of the crops is imperative.

Water is the most vital input in agriculture and has made a significant contribution in providing stability to food grain production and self-sufficiency. As compared to the
surface water, greater proportion of additional irrigation water comes from the ground water and this source is increasingly being exploited in an unscientific manner. It is estimated that around 88% of the fresh water resources are currently used for agriculture and remaining water is fulfilling the industrial and domestic requirements in the country. Presently, all the sectors of economy are demanding larger quantities of freshwater. Thus, tremendous amount of pressure lies on agriculture sector to reduce their share of water and at the same time to enhance total production, which could be achieved by enhancing productivity with increased water use efficiency, Heeman et. al [7]. In the global scenario, the demand for water has been on the rise in all water user sectors.

Sufficient efforts were not made in past to adopt efficient water distribution and application method. Many part of the irrigated areas have become water logged and affected by soil salinity problems and this has resulted into low productivity of fertile lands. This was mainly due to traditional flooding irrigation and slow adoption of scientific practices of irrigation water management. Added to these drawbacks, the major and medium irrigation projects also suffered from wide variations in soils, climate and cropping sequence, across the length and breadth of the irrigated command area. The present productivity of irrigated command areas around 2-3 tons per hectare as against 4 to 6 t/ha food grains demonstrated in research farms. The current food gain production of India is stagnating and population of the country is in increasing trend. Anticipating the crisis, the Government of India has always given priority to enhance irrigated area through various major, medium and minor irrigation and multipurpose projects, which were formulated and implemented through successive five-year plans. However, the judicious use of irrigation water for agriculture is equally important, productivity at the same time to save the irrigation water which is costly and limited resource. This can be achieved by introducing advanced methods of irrigation like Micro-irrigation coupled with other improved water management practices, Kumar [2] and Reddy [18].

2. MICRO IRRIGATION SYSTEMS
Micro irrigation system applies measured quantity of water slowly and directly above or below the soil surface, usually by discrete or continuous drops, tiny stream or miniature spray through emitters or applicators placed along a water delivery line near the plant.

2.1 Drip Irrigation System
Internationally, drip irrigation was developed originally as a sub-irrigation system and the basic idea underlying drip irrigation can be traced back to experiments in Germany in 1860s. The first work in drip irrigation in the United States was a study carried out by House in Colorado in 1913. An important break-through was made in Germany way back in 1920 when perforated pipe was introduced. During the early 1940s, Symcha Blass, an Israeli Engineer, observed that a big tree near a leaking pipe exhibited a more vigorous growth than other trees in the area, which were not reached by water from tap. This led him to the concept of an irrigation system that would apply water in small quantity literally drop by drop. In Israel, the first extensive research was conducted in the Arava and Negave deserts where adverse conditions of climate, very sandy alkaline soils and saline water had produced good results on crops grown with drip irrigation methods in comparison to conventional methods.

The drip irrigation was practiced in India since time immemorial for irrigating the religious Tulsi plant in many households during summer. With the development of plastic during and after Second World War, the idea of
using plastic pipe for irrigation becomes feasible. In the late 1940’s in the United Kingdom plastic pipe drip irrigation system was used to irrigate green houseplant. In 1971, the first International drip irrigation meeting was held in Tel Aviv, Israel, where 24 papers were presented. Drip irrigation in modern form was introduced in early 1971 in our country. Significant development has taken place in eighties, Narayananamoorthy [13]. Over the years, different companies developed different type of drippers. A cheap drip irrigation system was designed and fabricated at Tamil Nadu Agricultural University, Coimbatore in 1978, to find out the water used for various vegetables and fruit crops. The result indicated that the water required by crop in drip irrigation system was only 1/3rd to 1/4th of that required by surface irrigation method and the crop yield was to be higher in drip system.

Drip irrigation was practiced in India through indigenous methods such as bamboo pipes, perforated clay pipes and pitcher/porous cup irrigation. In Bamboo microirrigation systems, long hollow bamboo pipes of varying diameter (50-100 mm) are used for making channels. In Meghalaya, some of the farmers are using bamboo drip irrigation system for beetle, pepper and arecanut crops by diverting hill streams in hill slopes. The discharge at the head varies from 15 to 20 l/min, and is reduced to 10-30 drops/min, at the time of application. These methods can be advantageously used by individual farmers for smaller land holdings. In Maharashtra, perforated earthenware pipes were used and their efficiency and benefit cost ratio have been elaborated for popularizing them. Earthen pitchers and porous cups have also been used for growing vegetable crops in Rajasthan. The technique envisages embedding of earthen cups of 500 ml capacity at the site of seedlings. The cups are filled with water at an interval of 4 to 5 days.

2.2 Sprinkler Irrigation System
Under sprinkler irrigation water is sprinkled under pressure into the air and plant foliage through a set of nozzles attached to network of aluminium or High Density Poly Ethylene (HDPE) pipes in the form of rainfall, Christiansen [5]. These systems are suitable for irrigating crops where the plant density is very high where adoption of drip irrigation systems may not be economical. Sprinkler irrigation is suitable for horticultural crops like vegetables and seed spices. Conventionally, sprinkler irrigation has been widely in use for irrigating cereals, pulses, oil seeds and other field crops.

2.3 Crops Grown Under Drip and Sprinkler Irrigation System
Kumar [2] and Mane et. al. [12] tabulated and described that the following crops can be successfully grown under micro irrigation systems.

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Variety</th>
<th>Crops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Orchard crops</td>
<td>Grapes, banana, pomegranate, orange, citrus, tamarind, mango, fig, lemon, custard apple, sapota, guava, pineapple, coconut, cashew nut, papaya, aonla, litchi, water melon, musk melon</td>
</tr>
<tr>
<td>2.</td>
<td>Vegetables</td>
<td>Tomato, chilly, capsicum, cabbage, cauliflower, onion, okra, brinjal, bitter gourd, ridge gourd, cucumber, green peas, spinach, pumpkin, guard</td>
</tr>
</tbody>
</table>

Table 1: Crops Grown Under Drip and Sprinkler Irrigation System
2.4 Need of Design Software

The researches have developed the independent software for both drip design and sprinkler design systems, which gives the design parameters but none of these generate the design layout, which is equally important at field level. Andrade et. al. [1] and Hernandez et. al. [8] developed the software for sprinkler design, Rajput et. al. [17] developed the software for design of drip irrigation systems. The design and layout of drip and sprinkler irrigation systems consist of the collection of data from different sources with respect to various parameters to arrive at a feasible and practicable design for adoption in the field. It requires tremendous efforts and energy, planning, money to pool the information for the proper design of the systems and also, it is time consuming and cumbersome. Most of the designs require both qualitative and quantitative information to arrive at possible design from the reports and literature. In the developed software the above points have been addressed to the extent possible for drip and sprinkler design.

3. Overview of the Developed Software

The developed software has two modules – drip design module and sprinkler design module. The block diagram for the developed software may be shown as below:

![Block Diagram Of The Developed Software](image)

3.1 Steps in Design of Drip Irrigation System

The design of drip irrigation system differs from crop to crop, soil-to-soil and climatic conditions. The literature is very rich in providing the guidelines and design steps for drip design, [6], [9-12], [14-16]. In general following are the basic steps involved in design:

3.1.1 Selection of Drippers

The selection of dripper types and number of drippers per plant depends on peak water requirement of the crop, the crop and the soil. The emitter must supply enough water to the root zone of the crop to meet the crop water requirement. Normally, the emitters are located near the plant or the areas of high root concentration. Drip irrigation system can be designed as a point source or line source to supply water depending on the type of crops.

The required emitter flow rate can be calculated based on the water requirement of the crop, number of emitters per plant, irrigation application efficiency of the system and the duration of irrigation as:

\[
q_t = \frac{q_{req} + T}{E_a * e}
\]

(1)
Where,

- $q_r$ is the required emitter flow rate, l/h
- $q_t$ is the water requirement per plant per day, lit/plant/day
- $l$ is the irrigation interval, days
- $T$ is the irrigation time per set, hrs
- $E_a$ is the irrigation application efficiency
- $e$ is the number of emitter per plant

### 3.1.2 Selection and Design of Lateral

Lateral carries water from submain and feeds to the individual drippers. Generally, one lateral for each row or orchard plant and one lateral for two rows of sugarcane or vegetables are used. The size and length of lateral is decided by the discharge of the drippers and number of drippers on one lateral.

Calculation of lateral flow rate: Flow rate of lateral is the function of emitter discharge and number of emitter along the lateral,

$$Q_1 = K*n_e*q_a$$  \hspace{1cm} (2)

Where,

- $Q_1$ is the total flow rate of one lateral, l/h
- $n_e$ is the number of drippers on one lateral
- $q_a$ is the average dripper discharge, l/h
- $K$ is the constant, 1/3600

### 3.1.3 Selection and Design of Submain

Submain is generally made up of PVC (poly vinyl chloride) pipes of 32 mm, 40 mm, 50 mm, 63 mm and 75 mm in diameter. The design of submain is based on both capacity and uniformity. Capacity means the submain size should be large enough to deliver the required amount of water to irrigate the subsequent part of the field, Bralts et. al. [3].

Submain supplies water to individual lateral. Design of submain is similar to that of lateral, however it differs in that the spacing between outlets is greater and larger flow rates are involved. The size and length of submain is determined by number of laterals and distance between the laterals. Calculation of frictional head loss :

$$\Delta H_w = K \left(\frac{Q_{m}}{C}\right)^{1.852} D_{m}^{-0.871} L_{m} \times F$$  \hspace{1cm} (3)

Where,

- $\Delta H_w$ is the head loss in submain, m
- $K$ is constant, 1.21 x 10^{10}
- $Q_{m}$ is the flow rate in submain, l/sec
- $C$ is friction coefficient for continuous section of pipe depends on pipe material
- $D_{m}$ is the inside diameter of submain, mm
- $F$ is the outlet
- $L_{m}$ is the length of submain, m

### 3.1.4 Selection and Design of Mainline

Generally the size of mainline is one size higher than submain. The size of mainline is decided by flow rate of all submains. The sizes of mainline are 40 mm, 50 mm, 63 mm, 75 mm, 90 mm and 110 mm etc.

The head loss is calculated by (Using $C=150$),

$$\Delta H_w = 15.27 \left(\frac{Q_{m}}{D_{m}^{.871}}\right) L_{m}$$  \hspace{1cm} (4)

Where,

- $Q_{m}$ is the total discharge of mainline, l/sec
- $D_{m}$ is the inside diameter of mainline, cm
- $L_{m}$ is the length of mainline, m

OR

$$\Delta H_w = K \left(\frac{Q_{m}}{C}\right)^{1.852} D_{m}^{-0.871} L_{m}$$  \hspace{1cm} (5)

Where,

- $K$ is constant, 1.21 x 10^{10}
Software for design and layout of drip and sprinkler irrigation systems

C is the friction coefficient for continuous section of pipe and depends on pipe material

D\text{m} is the inside diameter of mainline, mm

### 3.1.5 Selection of Pump

Total head of pump = suction head + delivery head + filter losses + mainline losses + Operating pressure + fitting loss + ventury head loss + elevation difference

- Filter losses are assumed to be 2 m for screen filter (disc filter) and 2 m for sand filter
- Operating pressure is about 1 kg/cm\textsuperscript{2} (10 m)
- Fitting loss = 2 m
- Ventury head loss = 5 m

\[ \text{H.P.} = \left( \frac{Q.H.}{75 \times n_{\text{motor}} \times n_{\text{pump}}} \right) \]  \hspace{1cm} (6)

Where,
\[ Q \] is the maximum flow rate of system, l/sec
\[ H \] is the total head of the system
\[ n_{\text{motor}} \] is the motor efficiency, generally taken as 80%
\[ n_{\text{pump}} \] is the pump efficiency, generally taken as 75%
\[ \text{H.P.} \] is the horse power

### 3.1.6 Calculation of Irrigation Time

i) Water requirement (l/day/tree)

\[ \text{Irrigation time for tree crop (hrs)} = \frac{Q.H.}{D_{\text{gross}} f_i T} \]  \hspace{1cm} (7)

Where,
\[ Q \] is the discharge rate of dripper per tree (l/sec)
\[ H \] = gross depth of application, mm
\[ f_i \] = irrigation interval, days
\[ T \] = average actual operating time per day, h/day

ii) Water requirement per day (l/day/area)

\[ \text{Irrigation time for row crops (hrs)} = \frac{Q.H.}{S_s S_l} \]  \hspace{1cm} (8)

Where, discharge rate per unit area is calculated as lateral line discharge rate divided by row spacing.

### 3.2 Steps in Design of Sprinkler Irrigation System

Sprinklers spread water as “rain like” droplets over the land surface uniformly without runoff. Most sprinklers are either rotating or fixed-head type. The design details of sprinkler system components are explained by following steps:

#### 3.2.1 System Capacity Requirement

\[ Q_s = \frac{K A d_{\text{gross}}}{f_i T} \]  \hspace{1cm} (9)

Where,
\[ Q_s \] = system discharge capacity, l/s
\[ K \] = conversion constant, 2.78
\[ A \] = design area, ha
\[ D_{\text{gross}} \] = gross depth of application, mm
\[ f_i \] = irrigation interval, days
\[ T \] = average actual operating time per day, h/day

#### 3.2.2 Sprinkler Application Rate

\[ I = \frac{K Q}{S_s S_l} \]  \hspace{1cm} (11)

Where,
\[ I \] = average application rate
\[ K \] = conversion constant
\[ Q \] = sprinkler discharge
\[ S_s \] = spacing of sprinkler along the lateral, m
\[ S_l \] = spacing of lateral along the main line, m

#### 3.2.3 Nozzle Discharge and Pressure Relationship

\[ q = K_s \sqrt{H} \]  \hspace{1cm} (12)

Where,
\[ q \] = sprinkler discharge, l/min
\[ H \] = sprinkler operating pressure head, m
\[ K_s \] = appropriate discharge coefficient for the sprinkler and nozzle combined
3.2.4 Average Instantaneous Application Rate

\[ I_i = \frac{Kq}{\prod \left( R_j \right)^2 \left( S_a / 360 \right)} \]  
(13)

Where,

- \( I_i \) = average instantaneous application rate, mm/h
- \( K \) = conversion constant, 60
- \( q \) = sprinkler discharge, l/min
- \( R_j \) = radius of wetted area
- \( S_a \) = angular segment wetted by a stationary sprinkler jet, degrees

3.2.5 Uniformity Coefficient

\[ CU = 100 - 0.63 (100 - DU) \]

Where,

- \( CU \) = coefficient of uniformity, %
- \( DU \) = distribution uniformity, %

Average low quarter depth of water received

\[ DU = \frac{\text{Average low quarter depth of water received}}{\text{Average depth of water received}} \]
(14)

3.2.6 Fertigation Unit

India is the third largest fertilizer producing and consuming country in the world. The nutrient consumption per hectare and fertilizer use efficiency is very low in India. The main reason for the low efficiency is the type of fertilizer used and its method of application adopted by Indian farmers. Fertigation is the process of application of water-soluble solid fertilizer or liquid fertilizers through drip irrigation system. Through drip irrigation fertigation nutrients are applied directly into the wetted volume of soil immediately below the emitter where root activity is concentrated. Fertigation is possible in drip irrigation. Commonly used fertigation equipments are: venturi pumps (injector), fertilizer tank with flow by pass, pressure by pass tank and injection pumps.

3.2.7 Filtration Unit

Clogging of emitters is the most difficult problem encountered in drip irrigation system. The most common agents of clogging are the mineral and organic particles in the water source. Filtration of the water to check the contaminants entering the system is the best way against these occurrences. Clogged emitters are difficult to detect and expensive to clean or replace. The process of removing solid particles from liquid or gas by forcing through a porous medium is called as filtration. Setting basins, sand or media filters, screens, cartridge filter and centrifugal separators are the primary devices used to remove suspended material.

There are several types of commercially available filters, each based on different working principle and technologies and performs different filtration tasks according to water quality such as Gravel filters, Screen filters, Disc filters, Cartridge filters, Hydro cyclone filters.

4. DEVELOPED DESIGN SOFTWARE

As stated the developed software has two modules: 1) Design of drip irrigation system and 2) design of sprinkler irrigation system.
1671

4.1 Drip Design And Layout Module
The step by step flow diagram for the drip design module is shown in figure 2 and the Splash screen of the software is shown in Figure 3.

Main Screen of the Developed Software
The main screen of the software is shown in figure 4. There are two options for selection of required type of system - drip irrigation system and sprinkler irrigation system. One can select any of the system to proceed further based on requirement.

Basic Information for Drip Design
The design process starts with entering the basic information such as length and width of the plot, type of soil, slope of land, type of crop, water quality, type of shape of the plot etc. Just after entering and selection of values, next button is to be pressed to reach on the next screen.

Figure 2 : Flow Diagram For Drip Design Module
The steps for design of drip and sprinkler irrigation systems have been followed in chronological order, which are explained in previous pages. The final design is generated based on the input values and values entered or selected from various databases and respective boxes on the screens.

The two softwares used for developing of the design software are: Visual Basic as front end and Microsoft Access as back end. The developed software has many screens as per the steps followed for design of drip and sprinkler irrigation systems. The software starts with Splash screen followed by main screen, which gives the option for module selection.
Calculation of Crop Water Requirement

The next screen is for calculating the water requirement of the crop as described earlier. The input values on this screen are – name of crop, age of crop, crop coefficient, pan coefficient, row to row and plant-to-plant spacing etc. After selection and entering the values in various text and combo boxes, water requirement of the crop can be calculated by clicking on ‘Calculate’ button. The screen is shown in figure 6.

Calculation And Selection Of Lateral

Once the crop water requirement is calculated, next screen is for calculation and selection of diameter for lateral. Any diameter is selected from the dia list box and based on this, the head loss is calculated using Hizen and William equation. If head loss comes within the permissible limit i.e. 10% pressure variation then the lateral dia is acceptable. If not then next higher dia is selected and again the head loss is calculated, the same process is repeated till the calculated head loss is under permissible limit. The screen is shown in figure 7.

Design Layout Of Drip System

The design layout is generated based on the values selected or calculated in various screens. The design layout shows the information about the length, width, size and capacity of laterals, submain and mainline, distance and position of power source, capacity of the drippers, number of segments in the field, details of fertigation and filtration units required/selected etc. The screen is shown in figure 8.
Calculation Of Material Requirement

Finally, on the basis of generated design and layout, approximate cost of the complete drip system is calculated, which can be printed or output can be stored in a file. Before calculating the final system cost the filtration and fertigation equipment need be selected, otherwise software reminds that these units are to be selected before final cost of the is calculated. The screen is shown in figure 9.

4.2. Sprinkler Design And Layout Module

Sprinkler design is also based on the sequential steps starting from basic information to the final layout and cost estimates. The flow diagram for design of sprinkler irrigation system is shown in the figure 10

Screen For Basic Information For Sprinkler Design

Design starts with selection and entry of parameters from this screen. The basic information like plot size, crop type, overlap percentage, wind speed etc is entered or selected in this screen. The screen is shown in figure 11.
Based on the sprinkler size and water requirement, capacity of sprinkler system is calculated. Number of sprinklers is also calculated on this screen, the number is required for calculation of estimated cost. The screen is shown in figure 13.

Screen For Sprinkler Diameter Selection
The diameter of sprinkler pipe is selected based on the calculation for flow rate in the pipe. The screen is shown in figure 14.

Screen For The Generated Sprinkler Layout
Sprinkler layout is generated based on the values selected, entered or calculated on the previous screens. The generated design layout is shown in figure 15.

Screen For The Cost Estimate Of The Sprinkler System
After selection of filtration and fertigation units, cost estimate of the complete sprinkler system is calculated. The cost estimates can be viewed on the screen or a print out can be taken for future use. The screen is shown in figure 16.

5. CONCLUSION
The software has been developed keeping in view the actual requirement of the farmers, which has following features:
1. Provides the design of drip and sprinkler irrigation systems based on the selected parameters.
2. Material required and approximate cost estimate of the system is known in advance before installation the actual system in the field.
3. Various combinations of values can be tried as per the requirement of the user to get the suitable design.

REFERENCES


Author’s Biography

Dr. Konda Sreenivas Reddy, obtained the graduation degree in agricultural engineering from CAET, TNAU, Coimbatore, India in 1984, Master and PhD degree in Agricultural Engineering (SWCE) from Indian Agricultural Research Institute (IARI), New Delhi, India in 1990 and 1993 respectively. Worked as Assistant Professor at ANGRAU, Hyderabad from Aug 1993-Sept 1998. Worked as Senior Scientist (SWCE) at central Institute of Agricultural Engineering (CIAE), Bhopal from Sept 1998 to Sept 09,2006. Presently working as Principal Scientist (SWCE) at Central Institute for Dryland Agriculture( CRIDA), Hyderabad, India. Published two books, seven bulletins; four training manuals and 20 research papers. Presented 50 research papers in various conferences and symposia. Has the membership of the professional societies ISWAM, SCI, IASC, ISAE, IE(AP). Received ISAE team award during 2001-02 and Best M.Tech thesis award during 2002-03. Developed many useful and popular technologies related to SWCE during the service period. The area of research interests are Soil and Water Engineering.

Dr. Karan Singh Received the Bachelor of Science (B.Sc.) degree in 1986 from University of Allahabad, India and Master of Computer Applications (MCA) degree from Government Engineering College (RDVV), Jabalpur in 1993 and Doctor of Philosophy (PhD) degree in Computer Science in 2007 from Rani Durgawati Vishwavidyalaya (RDVV), Jabalpur, Madhya Pradesh, India. Joined as Scientist (Computer Applications in Agriculture) in 1997 through Agricultural Research Services (ARS). Presently, working as Senior Scientist (Computer Applications in Agriculture) at Central Institute of Agricultural Engineering, Bhopal, India. Published three research papers in international journals and seven research papers in reputed national journals. Also published one book and three technical bulletins. Has the membership of professional societies like Computer Society of India and Indian Society of Agricultural Engineers (ISAE).
ABSTRACT
This paper presents a new solution for processing the pulses from an optical encoder attached to a motor shaft, and deriving the rotational speed information. The measurement accuracy at low speeds using the proposed method is improved, compared to currently known methods. Another significant advantage of using the proposed circuit is that it can be implemented in specific hardware, thus, reserving the full computational power of the controlling microcontrollers for high-level control tasks and for future software expansions.

Keywords: Field Programmable Gate Array, Position Control, Quadrature Encoder/Decoder

1. INTRODUCTION
This paper describes the design of a high-performance circuit that performs both position and velocity measurements. A new solution for processing the pulses from the encoder and deriving the speed information is proposed. This is an adaptive technique allowing the evaluation of the speed with very good accuracy, even at very low rotational speeds. The main advantage of implementing the position and speed measurement circuitry in dedicated hardware is that the Microcontroller reserves its entire computational time for high-level control tasks. The proposed circuit is an interface between the microcontroller and the shaft quadrature encoder.

Simulation and experimental results from the designed and constructed chip are given.

All high-accuracy position control applications typically use an incremental or an absolute optical encoder mounted on the motor’s shaft. The output of the encoder usually provides quadrature pulses. These pulses are used to derive both position and angular velocity of the motor’s shaft. An ordinary incremental encoder does not give the absolute position but provides finer resolution. Many of the incremental encoders utilize quadrature sinusoidal scaling signals, which are generated by optical slits, for example. Typically, the rotating angle is measured by counting the number of sinusoidal waves or rectangular pulses, using a pulse counter.

The purpose of a quadrature decoder/counter is to take some of the real-time computational load off the microprocessor. When the processor reads the output signals of the position encoder, every change in state must be detected. This means that for an encoder of 250 pulses per channel per revolution turning at a modest 6000 rpm, the processor must detect and decode 100,000 state changes per second. This is difficult for most microcontrollers, and many systems use higher speeds and/or even denser encoders.
The solution to this difficulty is a circuit that will detect each of these state changes and send them to a counter. Each time a state change is detected in the positive direction, the decoder will increment the counter; a change in the other direction causes the counter to be decremented. Thus the counter will keep a running count of how far the encoder has moved, until it overflows. Now the microprocessor can read the number in the counter and compare that number to a previous reading in order to measure the distance traveled. Since the counter stores the position changes, a system using an 8-bit counter can measure the position from signals whose frequency is about 127 times higher than that of the fastest signals that can be measured by a software-only system. In a similar way, the 4-bit counter can read signals approximately 8 times faster than those read only by software. The typical application block diagram is shown in figure 1.

2. OPTICAL ENCODER / DECODER
Often used for digital position and velocity measurement
Two types - 1) absolute encoders: gives actual position
2) incremental encoders: gives change in position. Usually encoders’ measure angular displacement - can be used to measure rectilinear position - computer mouse (2-dimensional position) Incremental Encoder: a wheel with little windows (front and side views): as shown below

As wheel rotates, the photocell generates a digital signal as shown below figure.

If windows and dividers are of equal width, then the “on” and “off” times are of equal duration for constant rotation rate. Encoders actually generate a quadrature signal, consisting of two square waves that are 90 degree out of phase as shown below.

Figure 1 : Typical Application Block Diagram
Direction of rotation is determined by the phase difference between the two signals. For example, if Channel A leads Channel B, then the encoder is rotating (say) Clockwise direction (CW). If Channel B leads Channel A, then the direction of rotation is Counterclockwise direction (CCW). The direction of rotation can be found out by detecting the change of direction. The direction changing can be explained by using the signals below.

3. THE MICROCONTROLLER FOR MOTOR CONTROL
The classical approach of using a digital signal processor in position control is illustrated in Fig. 1a. This configuration consists of a position control loop that includes an inner speed control loop.

The angular rotor position is sampled and compared to the reference value. The position error is then processed by the position controller to provide the speed reference for the speed control loop. In the speed control loop, the motor speed is sampled and compared to the imposed reference value. Speed error is processed by the speed controller to provide an appropriate control signal for the motor drive.

In the commonly used configuration shown in Fig. 1(a), the microcontroller calculates the actual speed by making repeated arithmetic differentiation of the position information. In the proposed system [Fig. 1b], the microcontroller reads the actual position and velocity of the motor shaft directly from the proposed interface circuit, compares it with the desired position and speed, executes the control algorithm, and drives the motor.

4. POSITION MEASUREMENT
A dominant factor that limits the positioning accuracy of a servo controller is the angle resolution of an encoder attached to the motor shaft. The angle resolution of a conventional incremental encoder, available at reasonable cost, is 0.09. This resolution is sufficient for most applications, and the movement in high-speed positioning is also smooth. In low-speed operation, however, the intermittent movement due to digital control becomes appreciable, and the angle resolution of the encoder must be increased. Several methods for improving the resolution and accuracy of rotary encoders have been proposed in the literature. Among them, is an interpolation method the ways to improve the resolution of the available encoders, instead of using one having more precise scale, are: the multiply-by-four method, where 1/4 of the scale pitch resolution is obtained using a simple pulse technique; the interpolation of the scale interval of the currently used encoder the incremental encoder used in our application provides 1024 pulses/rotation. This limits
the position encoding accuracy to 360/1024=0.3520. In order to improve the angle resolution, the multiply-by-four technique is used. The multiply-by-four technique provides, in this case 360/4096=0.0870, angle resolution without additional cost to the application.

5. Velocity Measurement

It is based on counting encoder pulses in a specific time interval, and the corresponding time is measured by counting the clock pulses of a high-frequency clock with period Tc. The integer number of encoder pulses in a specific time interval. Both the pulse counter and the TC (time counter) are started at a rising edge of the encoder pulse. The counters are stopped by the first rising edge of the encoder pulse occurring after the interval Tc. The content of the pulse counter (Cp) is then the number of encoder pulses measured, and the content of the TC is the number of time clocks (Ct) measured. The values of Cp and Ct are used to calculate the rotational speed of the motor since is the angular position difference and is the accurate sampling time of each measurement.

6. Functional Description

A. Digital Filter

The digital filter is responsible for rejecting noise on the incoming quadrature signals. This is achieved by combining Schmitt triggered inputs and three clock-cycle delay filters. This combination rejects low level noise and large, short duration noise spikes that typically occur in motor system applications. The signals on each channel are sampled on rising clock edges. A time history of the signals is stored in the four-bit shift register. Any change on the input is tested for a stable level being present for three consecutive rising clock edges. Therefore, the filtered output waveforms can change only after an input level has the same value for three consecutive rising clock edges. By this method the short noise spikes between rising clock edges are ignored and pulses shorter than two-clock period are rejected.

B. Decoder Logic

The decoder logic decodes the incoming filtered signals into count information. The decoder samples the outputs of channels A and B. Based on the past binary state of the two signals and the present state, it outputs a count signal (CNT) and direction signal (UP_DN) to the position counter, see state transition diagram below. Channel A leading Channel B results in counting up. Channel B leading channel A results in counting down.

Based on the states of the motor rotor position and quadrature signal came from the encoder the direction of the rotation of the motor is find out by the decoder logic the signal and states of each rotor position is as shown in the signal diagram f and logic for each state change is shown in the table below.
C. Position Counter

Table 1: Direction Decoding

Channel A leading channel B results in counting up.
Channel B leading Channel A results in count down.

Table 2: States of Motor

<table>
<thead>
<tr>
<th>CH A</th>
<th>CH B</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>S1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>S2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>S3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>S4</td>
</tr>
</tbody>
</table>

The binary up/down counters count on rising clock edge. Channel A leading channel B results in counting up. Channel B leading Channel A results in count down.

D. Position Data Latch

The position data latch captures the position counter output data on each rising clock edge, except when its inputs are disabled by the inhibit logic during two-byte operation. The inhibit logic samples the OE and SEL control signals and does the conditional inhibit of the position data register.

E. Bus I/F Logic

Consists of multiplexers and tri-state output buffer. This allows the independent access to the low and high bytes of the position data latch.

7. Testing and Experimental Results

The simulation result of the quadrature decoder/counter is shown below. The simulation is done by using ModelSim Simulator.
8. CONCLUSION

In this paper, an incremental encoder based position and velocity measurement FPGA with SPI interface has been described. It is based on two techniques: counting pulses from a clock between successive pulses of the encoder for low speed application and counting the number of pulses from the encoder in a known time for medium and high speeds. The combined techniques enable the chip to deliver high accuracy for a wide speed range without degradation in dynamic behavior. With on-chip velocity estimate, it reduces computational complexity imposed on the supervisory microcontroller.

REFERENCES


Author’s Biography

Anitha Mary.X completed Bachelor of Engineering in Electronics and Instrumentation in Karunya Institute of Technology, Coimbatore and Master of Engineering in Anna University, Coimbatore. Currently pursuing Ph.D in Control System in Karunya University. She awarded Bharathiar University 7th Rank in U.G and currently
FPGA Based Quadrature Decoder/Counter with I²C Bus Interface for Motor Control

guiding many U.G. and P.G. Area of Interest is Control system, FPGA. Thilagavathy is working as lecturer in Karunya University

Manu is a P.G. Scholar from Karunya University, Coimbatore.
ABSTRACT

The design of a n-bit binary parallel adder for a VLSI circuit is trade-off between speed of operation and hardware complexity (chip area). When speed is not the concern, Ripple Carry Adder (RCA) is the best choice, because it occupies less area and has a regular structure. The delay is directly proportional to the number of bits of the adder. Today performance is much more important than chip area; the Carry Look Ahead (CLA) adder may be the right choice in which the carry bits are predicted well in advance to speed up the computation. But when the number of bits increases, the fan-in and fan-out of the CLA increases and hence speed starts to deteriorate. In order to compensate this drawback parallel prefix adders are preferred. Parallel prefix adders are slight variation of CLA in which the carry bits are generated parallely without increasing the fan-in and fan-out of the computation nodes to a larger extent. The proposed PPA structure has given better optimization in power consumption, delay and power delay product. A comparative study is made between the proposed structure and existing PPA algorithms. All the circuits were simulated using Tanner EDA in 180nm technology.

Keywords: Parallel Prefix Adder, Dot Operator, Semi-Dot Operator.

1. INTRODUCTION

VLSI Integer adders find applications in Arithmetic and Logic units (ALU’s), microprocessors and memory addressing units. Speed of the adder often decides the minimum clock cycle time in a microprocessor. The need for a Parallel Prefix adder (PPA) is that it is primarily fast when compared with a ripple carry adder. PPA is family of adders derived from the commonly known carry look ahead adders. These adders are suited for adders with wider word lengths. PPA circuits use a tree network to reduce the latency to $O(\log_2 n)$ where ‘n’ represents the number of bits. This chapter discusses the design proposal of new prefix adder architecture for 8-bit, 16-bit and 32-bit addition. The proposed architectures have the least number of computation nodes when compared with its peer existing one’s. This reduction in hardware of the proposed architectures helps to reap a benefit in the form of least power and least power delay product.

2. EXISTING PARALLEL PREFIX ADDERS

The structure of the prefix network specifies the type of the PPA. The Prefix network described by Haiku Zhu, Chung-Kuan Cheng and Ronald Graham [1], has the minimal depth for a given ‘n’ bit adder. Optimal logarithmic adder structures with a fan-out of two for minimizing the area-delay product is presented by Matthew Ziegler and Mircea Stan [2]. The Sklansky adder [3] presents a minimum depth prefix network at the cost of increased fan-out for certain computation nodes. In Sklansky adder, the fan-out from the inputs to outputs along the critical path increases drastically which...
Hybrid Parallel Prefix adder for High Performance Computing

introduces latency in the circuit. The algorithm invented by Kogge-Stone [4] has both optimal depth and low fan-out but produces massively complex circuit realizations and also account for large number of interconnects. Kogge-Stone adder possesses a regular layout and is preferred for high performance applications. Brent-Kung adder [5] has the merit of minimal number of computation nodes, which yields in reduced area but structure has maximum depth which yields slight increase in latency when compared with other structures. Brent Kung adder is oriented towards simpler tree structure with a fewer computation nodes. The Han-Carlson adder [6] combines Brent-Kung and Kogge-Stone structures to achieve a balance between logic depth and interconnect count. Han-Carlson adder the reduces the hardware complexity when compared to that of a Kogge-Stone adder but at the cost of introducing a additional stage to its carry merge path. Ladner and Fischer [7] proposed a general method to construct a prefix network with slightly higher depth when compared with Sklansky topology but achieved some merit by reducing the maximum fan-out for computation nodes in the critical path. Fischer adder is an improved version of Sklansky adder, where the maximum fan-out is reduced. Taxonomy of classical Prefix Parallel Adders based on fan-out, interconnect count and depth characteristics has been presented by Harris [8]. In this paper, a novel hybrid prefix adder structure for 8-bit, 16-bit and 32-bit has been proposed. The proposed structures have the least power delay product amongst all its peer one’s.

3. COMPARATIVE STUDY ON EXISTING PREFIX ADDERS

Table 1 summarizes the data regarding the requirement of number of computation nodes and logic depth for various existing Parallel Prefix adders. Let ‘n’ be the word-length of the adder in terms of bits.

<table>
<thead>
<tr>
<th>Adder Type</th>
<th>Number of Computation Nodes</th>
<th>Logic Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent-Kung</td>
<td>[2 - \log_2 n - 2]</td>
<td>[2 - \log_2 n - 2]</td>
</tr>
<tr>
<td>Kogge-Stone</td>
<td>(\lceil\log_2 n\rceil)</td>
<td>(\lceil\log_2 n\rceil)</td>
</tr>
<tr>
<td>Han-Carlson</td>
<td>(\frac{n}{2}\log_2 n)</td>
<td>(\frac{n}{2}\log_2 n)</td>
</tr>
<tr>
<td>Ladner-Fischer</td>
<td>(\frac{n}{2}\log_2 n)</td>
<td>(\frac{n}{2}\log_2 n)</td>
</tr>
<tr>
<td>Sklansky</td>
<td>(\frac{n}{2}\log_2 n)</td>
<td>(\frac{n}{2}\log_2 n)</td>
</tr>
</tbody>
</table>

4. PROPOSED HYBRID PREFIX ADDER ARCHITECTURE

The Proposed 32-bit Parallel Prefix adder architectures is shown in Figure.1. The architectures employ the associative property of the PPA to keep the number of computation nodes at a minimum, by eliminating the massive overlap between the prefix sub-terms being computed.

The Proposed adder structures are implemented using three schemes namely

1) Scheme I
2) Scheme II
3) Scheme III
4.1. Properties of Dot Operator

The prefix operator (•) has two essential properties which allow for greater parallelism.

1) Associative property is listed in equation (1) and (2) given below

\[
(G, \bar{K})_{[h,k]} \cdot (G, \bar{K})_{[j,k]} = (G, \bar{K})_{[h,j]}
\]  
\[
(G, \bar{K})_{[h,j]} \cdot (G, \bar{K})_{[i,k]} = (G, \bar{K})_{[h,i]}
\]

Where \( h \geq j \geq k \).

2) Idempotent property is listed in equation (3) and (4) respectively

\[
(G, \bar{K})_{[i,j]} \cdot (G, \bar{K})_{[j,k]} = (G, \bar{K})_{[i,k]}
\]  
\[
(G, \bar{K})_{[i,j]} \cdot (G, \bar{K})_{[j,k]} = (G, \bar{K})_{[i,j]}
\]

Associativity allows pre-computation of sub-terms of the prefix equations. This indicates that a serial iteration implied by the above prefix operation can be parallelized. Idempotency allows these sub-terms to overlap, which provides some useful flexibility in the parallelization.

4.2. Scheme I

The first stage of the computation is called as preprocessing. The first stage in the architectures of the proposed 32-bit prefix adder involve the creation of complementary generate and propagate signals for individual operand bits in active low format. The equations (5) and (6) represent the functionality of the first stage.

\[
\bar{G}_i = (a_i \cdot \bar{b}_i)
\]  
\[
\bar{P}_i = a_i \oplus b_i = a_i \cdot \bar{b}_i
\]

In the above equations, \( a_i, b_i \) represent input operand bits for the adder, where \( i \) varies from 0 to 31. The second stage in the Prefix addition is termed as prefix computation. This stage is responsible for creation of group generates and groups propagate signals. For deriving the carry signals in the second stage, this architecture introduces four different computation nodes for achieving improved performance. There are two cells designed for the dot operator. First cell for the dot operator named odd-dot represented by a ‘•’, works with active low inputs and generates active high outputs.

The second cell for the dot operator named even-dot represented by a ‘•’, works with active high inputs and generates active low outputs. Similarly, there are two cells designed for the semi-dot operator. First cell for the semi-dot operator named odd-semi-dot represented by a ‘•’, works with active low inputs and generates active high outputs. The second cell for the semi-dot operator named even-semi-dot represented by a ‘•’, works with active high inputs and generates active low outputs. The last computation node in each column of the architecture is a semi-dot operator. The stages with odd indexes use odd-dot and odd-semi-dot cells where as the stages with even indexes use even-dot and even-semi-dot cells. The equations (7) and (8) represent the functionality of odd-dot and even-dot cells respectively.

\[
((G, \bar{P})_{[i,j]} \cdot (G, \bar{P})_{[j,k]}) = (G_{[i,j]} + P_{[i,j]} \cdot G_{[j,k]} + P_{[j,k]} \cdot P_{[i,k]})
\]

\[
((G, \bar{P})_{[i,j]} \cdot (G, \bar{P})_{[j,k]}) = (G_{[i,j]} + P_{[i,j]} \cdot G_{[j,k]} + P_{[j,k]} \cdot P_{[i,k]})
\]

The equations (9) and (10) represent the functionality of odd-semi-dot and even-semi-dot cells respectively.
Hybrid Parallel Prefix adder for High Performance Computing

\[
(G_{i,0}) = (\overline{G}, \overline{P})_{i,j} \cdot (G, P)_{j+1,i} \\
= ((G_{i,j}, \overline{P}_{i,j}) + G_{i,j+1}) \\
= (G_{i,j} + P_{i,j}) = c_i
\]  
(9)

\[
(\overline{G}_{i,0}) = (G, P)_{i,j} \cdot G_{i,j+1} \\
= (G_{i,j} + P_{i,j} \overline{G}_{j+1,i}) = c_i
\]  
(10)

CMOS logic family will implement only inverting functions. The inverting property of CMOS logic is employed. If both the inputs of a computation node in stage ‘i’ are from stage ‘(2 * i – 1)’, then an alternative cascade of odd computation cell and even computation cell derives the benefit of elimination of two pairs of inverters between successive stages for each computation node in stage ‘i’. If a dot or a semi-dot computation node in a stage ‘i’ receives any of its inputs from stage ‘(2 * i – 1)’, then it is essential to introduce a pair of inverters in a path. From the prefix graph of the proposed structure shown in Fig. 1, it is observed that there are only few edges with a pair of inverters, to make \((G, P)\) or to make \((\overline{G}, \overline{P})\) respectively. The pair of inverters in a path is represented by a ‘\(\square\)’ in the prefix graph. Thus by introducing two cells for dot operator and two cells for semi-dot operator, we have eliminated a large number of inverters. Due to inverter elimination in paths, the propagation delay in those paths would have reduced. Further it accounts for a benefit in power reduction, since these inverters if not eliminated, would have contributed to significant amount of power dissipation due to switching.

The output of the odd-semi-dot cells gives the value of the carry signal in that corresponding bit position. The output of the even-semi-dot cell gives the complemented value of carry signal in that corresponding bit position.

The final stage in the prefix addition is termed as post-processing. The final stage involves generation of sum bits from the active low Propagate signals of the individual operand bits and the carry bits generated in true form or complement form. The proposed 32-bit structure has a maximum fan-out of 6 and a logic depth of 9. The lateral fan-out slightly increases, but we get an advantage of limited interconnect lengths since the prefix graph grows along the main diagonal.

4.3. Scheme II

The first stage in the architectures of the proposed prefix adder structures involves the creation of kill and complementary generate and propagate for individual operand bits using the equations (11), (12) and (13) respectively

\[
K = a_i + b_i
\]  
(11)

\[
G_i = (a_i \cdot b_i)
\]  
(12)

\[
\overline{P} = (a_i \oplus b_i)
\]  
(13)

In the above equations, \(a_i, b_i\) represent input operand bits for the adder, where ‘i’ varies from 0 to 7 for 8-bit, 0 to 15 for 16-bit and 0 to 31 for 32-bit adders respectively. For deriving the carry signals in the second stage, this architecture introduces four different computation nodes for achieving improved performance. There are two cells designed for the dot operator. First cell for the dot operator named odd-dot represented by a ‘\(\square\)’, is defined by the equation (14)

\[
K = a_i + b_i
\]  
(14)

The second cell for the dot operator named even-dot represented by a ‘\(\square\)’, is defined by the equation (15)
Similarly, there are two cells designed for the semi-dot operator. First cell for the semi-dot operator named odd-semi-dot represented by a ‘$\cdot$’, the second cell for the semi-dot operator named even-semi-dot represented by a ‘$\cdot\ast$’, works are defined using equations (16) and (17) respectively.

$$$(G, K)_{[i,j]} = (G_{[i,j]} \ast (G, K)_{[j,i+1]})$$ (15)$$

The stages with odd indexes use odd-dot and odd-semi-dot cells while as the stages with even indexes use even-dot and even-semi-dot cells. CMOS logic family will implement only inverting functions. Thus cascading odd cells and even cells alternatively gives the benefit of elimination of two inverters between them, if a dot or a semi-dot computation node in an odd stage receives both of its input edges from any of the even stages and vice-versa. But it is essential to introduce two inverters in a path, if a dot or a semi-dot computation node in an even stage receives any of its edges from any of the even stages and vice-versa. From the prefix graph of the proposed structures, we observe that there are only few edges with a pair of inverters, to make $(G, K)$ as $(G, K)$ or to make $(G, K)$ as $(G, K)$ respectively. The pair of inverters in a path is represented by a ‘$\ast$’ in the prefix graph. By introducing two cells for dot operator and two cells for semi-dot operator, we have eliminated a large number of inverters. Due to inverter elimination in paths, the propagation delay in these paths has reduced. Further we achieve a benefit in power reduction, since these inverters if not eliminated, would have contributed to significant amount of power dissipation due to switching. The output of the odd-semi-dot cells gives the value of the carry signal in that corresponding bit position. The output of the even-semi-dot cell gives the complemented value of carry signal in that corresponding bit position. The final stage involves generation of sum bits from the Propagate signals of the individual operand bits and the carry bits generated in true form or complement form.

4.4 Scheme III

This scheme is a slight variation of Scheme II. The first stage involves creation of kill and complementary generate signal only for the individual operand bits. The Propagate signal is derived from the kill and the generate signal using a NOR operation. The rest of the architecture is similar to Scheme II.

5. SIMULATION ENVIRONMENT

Simulation for the Parallel Prefix adder designs was done using Tanner EDA in 180nm technology. All the Parallel Prefix Adder structures were implemented using CMOS logic family. The aspect ratio of the MOS transistor devices were chosen such that $W_L = 3*W_L$. For TSMC 180 nm technology, threshold voltages of NMOS and PMOS transistors are around 0.3694 V and -0.3944 V respectively and the supply voltage was kept at 1.8 V. The rise time and fall times of the input waveforms were set to 0.10 ns. The parameters considered for comparison are power consumption, worst case delay and power-delay product. The various PPA structures
were then compared with the number of computation nodes needed for circuit realizations. The input patterns were switched after every 10 ns.

### 6. Simulation Results and Analysis

Table 2 lists the structural characteristics of various 32-bit Parallel Prefix adders. From this table it is observed that the Proposed 32-bit Parallel Prefix adders have the least number of computation nodes amongst all other peer designs.

<table>
<thead>
<tr>
<th>Adder Name</th>
<th>Number of Computation Nodes</th>
<th>Logic Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent-Keng</td>
<td>26</td>
<td>8</td>
</tr>
<tr>
<td>Kogge-Stone</td>
<td>98</td>
<td>5</td>
</tr>
<tr>
<td>Han-Carlson</td>
<td>33</td>
<td>6</td>
</tr>
<tr>
<td>Ladner-Fischer</td>
<td>33</td>
<td>6</td>
</tr>
<tr>
<td>Sklansky</td>
<td>33</td>
<td>5</td>
</tr>
<tr>
<td>Proposed</td>
<td>23</td>
<td>9</td>
</tr>
</tbody>
</table>

#### 6.1 Simulation Results for Proposed Hybrid Parallel Prefix Adders Using Scheme I

Table 3: Performance Comparison of 32-bit Parallel Prefix Adders

<table>
<thead>
<tr>
<th>Adder Name</th>
<th>Average Power (μW)</th>
<th>Delay (ns)</th>
<th>Power-Delay Product (X 10^6 Joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent-Keng</td>
<td>211.6303</td>
<td>1.05</td>
<td>222.211815</td>
</tr>
<tr>
<td>Kogge-Stone</td>
<td>352.5317</td>
<td>0.79</td>
<td>278.500043</td>
</tr>
<tr>
<td>Han-Carlson</td>
<td>258.2679</td>
<td>0.89</td>
<td>212.859081</td>
</tr>
<tr>
<td>Sklansky</td>
<td>272.2407</td>
<td>0.70</td>
<td>190.56849</td>
</tr>
<tr>
<td>Ladner-Fischer</td>
<td>217.3403</td>
<td>0.91</td>
<td>197.779673</td>
</tr>
<tr>
<td>Proposed</td>
<td>210.8441</td>
<td>0.85</td>
<td>179.382485</td>
</tr>
</tbody>
</table>

#### 6.2. Simulation Results for Proposed Hybrid Parallel Prefix Adders Using Scheme II

<table>
<thead>
<tr>
<th>Adder Name</th>
<th>Average Power (μW)</th>
<th>Delay (ns)</th>
<th>Power-Delay Product (X 10^6 Joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent-Keng</td>
<td>197.0751</td>
<td>1.39</td>
<td>273.93489</td>
</tr>
<tr>
<td>Kogge-Stone</td>
<td>580.2907</td>
<td>0.92</td>
<td>322.67444</td>
</tr>
<tr>
<td>Han-Carlson</td>
<td>233.8426</td>
<td>1.25</td>
<td>292.30225</td>
</tr>
<tr>
<td>Sklansky</td>
<td>263.7564</td>
<td>1.11</td>
<td>292.747404</td>
</tr>
<tr>
<td>Ladner-Fischer</td>
<td>211.5786</td>
<td>1.23</td>
<td>260.364678</td>
</tr>
<tr>
<td>Proposed</td>
<td>204.261</td>
<td>1.25</td>
<td>255.32635</td>
</tr>
</tbody>
</table>

#### 6.3. Simulation Results for Proposed Hybrid Parallel Prefix Adders Using Scheme III

<table>
<thead>
<tr>
<th>Adder Name</th>
<th>Average Power (μW)</th>
<th>Delay (ns)</th>
<th>Power-Delay Product (X 10^6 Joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brent-Keng</td>
<td>196.4073</td>
<td>1.42</td>
<td>319.714495</td>
</tr>
<tr>
<td>Kogge-Stone</td>
<td>336.8599</td>
<td>0.96</td>
<td>325.40064</td>
</tr>
<tr>
<td>Han-Carlson</td>
<td>220.71358</td>
<td>1.13</td>
<td>249.40635</td>
</tr>
<tr>
<td>Sklansky</td>
<td>242.175</td>
<td>1.07</td>
<td>261.26725</td>
</tr>
<tr>
<td>Ladner-Fischer</td>
<td>207.5353</td>
<td>1.25</td>
<td>259.419125</td>
</tr>
<tr>
<td>Proposed</td>
<td>187.0996</td>
<td>1.14</td>
<td>213.20328</td>
</tr>
</tbody>
</table>

Figure 2: Power Comparison of 32-bit Adders In All Three Schemes
Figure 3: Power Delay Product Comparison Of 32-bit Adders In All Three Schemes

7. CONCLUSION

Thus from the analysis it is clear that the power consumption and the power delay product of the proposed structure is found to be reduced. An improvement of 29% to 36% has been achieved in power delay product over Brent Kung Adder and 29% to 43% in power over Kogge Stone Adder. It is inferred that the power saving gradually increases for the proposed architecture implemented using Scheme III when the size of the prefix adder grows. It is also observed that the power delay product of the proposed architecture is minimal when the word length of the adder increases. The proposed architecture implemented using scheme III design consumes least power with very less delay penalty when compared with its peer existing prefix adders. The Proposed Adder has a distinct advantage not only in terms of speed of performance, but also regarding gate count when compared to other adders. Thus it can be used in DSP, memory addressing and microcontroller applications which demands fast addition.

REFERENCES


