DYNAMIC ROUTING WITH SECURITY CONSIDERATION

PROJECT REPORT

Submitted By

G.PUSHPARANI

Register No.: 0720300031

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

in

COMPUTER APPLICATIONS

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

May, 2010

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 006.

Department of Computer Applications

PROJECT WORK

MAY 2010

This is to certify that the project entitled

DYNAMIC ROUTING WITH SECURITY CONSIDERATION

is the bonafide record of project work done by

G.PUSHPARANI

Register No: 0720300031

of MCA (Computer Applications) during the year 2009-2010.

06\05\10

Project Guide

Head of the Department

Submitted for the Project Viva-Voce examination held on 17.05.2010

17/5/10

17/5/10

DECLARATION

I affirm that the project work titled **DYNAMIC ROUTING WITH SECURITY CONSIDERATION** being submitted in partial fulfilment for the award of **MASTER OF COMPUTER APPLICATIONS** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

G.PUSHPARANI,

Register No.: 0720300031.

I certify that the declaration made above by the candidate is true

Signature of the Guide,

With Name & Designation

[V. JARAJA JAYA CHITH

LECTURER]

Maa-yoke
ASSURING THE AZURE

MAA YOKE TECHNOLOGY

14.05.10

PROJECT COMPLETION CERTIFICATE

This is to certify that the project work entitled

"DYNAMIC ROUTING WITH SECURITY CONSIDERATION"

Is a Bonafide work done

by

G.Pushparani (Reg.No: 0720300031)

Of Final year MCA

Student of
Kumaraguru College of Technology,
Coimbatore.

She has undergone her project work from
Dec'09 to May '10
and completed her Project successfully.

During the period of the project her performance was **GOOD.**

Asha .S

Project Incharge

MAA-YOKE TECHNOLOGY
CHENNAI
600 015

ABSTRACT

Security has become one of the major issues for data communication over wired and wireless networks. Different from the past work on the designs of cryptography algorithms and system infrastructures, we will propose a dynamic routing algorithm that could randomize delivery paths for data transmission.

The algorithm is easy to implement and compatible with popular routing protocols, such as the Routing Information Protocol in wired networks and Destination-Sequenced Distance Vector protocol in wireless networks, without introducing extra control messages. An analytic study on the proposed algorithm is presented, and a series of simulation experiments are conducted to verify the analytic results and to show the capability of the proposed algorithm.

The objective of this work is to explore a security enhanced dynamic routing algorithm based on distributed routing information widely supported in existing wired and wireless networks. We aim at the randomization of delivery paths for data transmission to provide considerably small path similarity (i.e., the number of common links between two delivery paths) of two consecutive transmitted packets.

The project will be developed with C#. NET.

ACKNOWLEDGEMENT

I wish to express my deep unfathomable feeling of gratitude and indebtedness to **Dr. S. Ramachandran, Principal,** Kumaraguru College of Technology, and Coimbatore for the successful completion of the project work.

I am very glad to express a special word of thanks to **Dr. A. Muthukumar, Professor and Course Coordinator,** Department of Computer Applications, Kumaraguru College of Technology, Coimbatore for encouraging me to do this work.

I am very much indebted to **Mr. S. Hameed Ibrahim, Senior lecturer,** Project Coordinator, Department of Computer Applications, Kumaraguru College of Technology, and Coimbatore for her complete assistance, guidance and support given to me throughout my project.

I am very much indebted to my guide **Ms. V. JALAJA JAYALAKSHMI, Lecturer,** Department of Computer Applications, Kumaraguru College of Technology, Coimbatore for his complete assistance, guidance and support given to me throughout my project.

Finally, I owe a lot to my beloved parents and family members and to my friends and to my department staffs for their help and co-operation to complete this project successfully.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE ORGANIZATION

MAA-YOKE possesses consultants who have earned the respect of Clients because our consultants are seasoned veterans of the IT world who have consistently delivered over time. We pride ourselves on our ability to get the job done on time, on Budget and above all meeting the expectations of our customers. Our Services are focused around the delivery and management of large scale, complex IT infrastructures. Our resources, including our senior staff, have worked with large complex clients. Our experience spans the areas from project management to highly specific technical skills as required by our clients.

MAA-YOKE TECHNOLOGY is an organization created by experienced business partners, who bring together a wide variety of experience in the information technology field IT planning, Project Management, Operations Management, Marketing and Finance with the expertise to deal with both public and private sectors. Through its partners, staff and strategic alliances, MAA-YOKE brings to its clients a grouping of in-depth skills and abilities providing Project and Implementation Services that strive to maintain and reduce the Total Cost of Ownership for Infrastructure Technology.

MAA-YOKE has a great deal of experience and knowledge and can assist organizations in the following areas:

IT Service Management
Engineering
IT Sales, Service & Support

OUR MISSION

Our mission is to assist our clients in attaining the lowest practical total cost of ownership (TCO) by streamlining the deployment of human resources to cater to the requirements of current and emerging technologies.

OUR VALUES

One of the fundamental principles at MAA-YOKE is our desire to provide our customers with a reliable service based on experience and performance to meet the imposing standards of delivery. We are an achievement-oriented professional services company, and we subscribe to the disciplines and goals of technology and business management principles.

HIGHLIGHTS

MAA-YOKE's teams of employees and senior management have been at the forefront of the delivery and support of information technology services.

OUR EXPERIENCE:-

MAA-YOKE Technology is an IT consulting firm that supports clients with large information technology infrastructures. We specialize in the development and implementation of office productivity technology management plans for managing LAN & WAN based information technology services and infrastructure used in the daily business operations of our clients.

MAA-YOKE's workforce has the knowledge and experience to help clients develop and implement best practice leadership within their company's environment. Client prioritization often benefits from our analysis of operating costs, development costs, service availability demands, and fiscal resource requirements to support a robust computing.

Our support in the following fields have been appreciated among the IT fraternity

- IBM TECHNOLOGIES
- MICROSOFT TECHNOLOGIES
- SUN MICROSYSTEMS
- ORACLE CORPORATION
- LINUX / UNIX TECHNOLOGIES
- ERP PACKAGES
- DATAWAREHOUSING
- EMBEDDED SYSTEMS
- NETWORK SECURITY

1.2 ABOUT THE PROJECT

In the past decades, various security-enhanced measures have been proposed to improve the security of data transmission over public networks. Existing work on security-enhanced data transmission includes the designs of cryptography algorithms and system infrastructures and security-enhanced routing methods. Their common objectives are often to defeat various threats over the Internet, including eavesdropping, spoofing, session hijacking, etc. Among many well-known designs for cryptography based systems, the IP Security (IPSec) and the Secure Socket Layer (SSL) are popularly supported and implemented in many systems and platforms. Although IPSec and SSL do greatly improve the security level for data transmission, they unavoidably introduce substantial overheads especially on gateway/host performance and effective network bandwidth.

For example, the data transmission overhead is 5 cycles/byte over an Intel Pentium II with the Linux IP stack alone, and the overhead increases to 58 cycles/byte when Advanced Encryption Standard (AES) is adopted for encryption/decryption for IPSec. Another alternative for security-enhanced data transmission is to dynamically route packets between each source and its destination so that the chance for system break-in, due to successful interception of consecutive packets for a session, is slim. The intention of security-enhanced routing is different from the adopting of multiple paths between a source and a destination to increase the throughput of data transmission.

The objective of this work is to explore a security enhanced dynamic routing algorithm based on distributed routing information widely supported in existing wired and wireless networks. We aim at the randomization of delivery paths for data transmission to provide considerably small path similarity (i.e., the number of common links between two delivery paths) of two consecutive transmitted packets.

The proposed algorithm should be easy to implement and compatible with popular routing protocols, such as the Routing Information Protocol (RIP) for wired networks and Destination-Sequenced Distance Vector (DSDV) protocol for wireless networks, over existing infrastructures.

SYSTEM ANALYSIS

CHAPTER 2

SYSTEM ANALYSIS

2.1 INTRODUCTION

System analysis is the second phase of System Development Life Cycle. Here we are analyzing all the requirements needed to develop a project. For this we are conducting Feasibility Study and Requirements analysis. The first and most important job of this phase is requirement determination. Finally, the requirements are studied and structured in accordance with their inter-relationships.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is a problem solving activity that requires communication between the system users and the system developers. It does various feasibility studies through which a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study.

2.2 EXISTING SYSTEM

The system, which is followed at present, is a manual system. Important drawback of existing system is cost, distance and time factor. It will not help the management to solve the problem in time.

Drawbacks of the Existing System:

- The existing system consumes more time and more cost.
- Security problem.
- In existing system algorithm are different for wired and wireless network.
- For wired network we use Routing Information protocol.
- For wireless network we use Destination Sequence Distance Vector.

The common objectives are

1. To defeat threats.
2. Evas dropping
3. Spoofing.
4. Session hijacking.

Existing system uses

1. Ipsec.
2. Secured Socket layer.

2.3 PROPOSED SYSTEM

The main objective of the proposed system is to reduce man power and distance, cost conception. It is more user- friendly, fast, efficient and accurate. The limitation of the existing system can be overcome by the proposed system.

Advantages of Proposed System:

- An alternative method is proposed to maintain security without overhead.
- By dynamically routing packet.
- But this needs more number of packet transfer which leads to

  1) Manage Flooding.

  2) Extra control message is passed which may increase traffic.

2.4 FEASIBILITY STUDY

All projects are feasible given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. Feasibility and risk analysis is related in many ways. If project risk is great , the feasibility listed below is equally important.

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

ECONOMICAL FEASIBILITY

Economic feasibility is the most frequently used method for evaluating the effectiveness of a candidate system. More commonly known as cost / benefit analysis, the procedure is to determine the benefits and saving that are expected from a candidate and compare them with the costs. If the benefits outweigh cost. Then the decision is made to design and implement the system. Otherwise drop the system.

This system has been implemented such that it can be used to analysis the traffic. So it does not require any extra equipment or hardware to implement. So it is economically feasible to use.

TECHNICAL FEASIBILITY

Technical feasibility centers on the existing computer system (hardware, software, etc..) and to what extent it can support the proposed addition. For example, if the current computer is operating at 80% capacity. This involves, additional hardware (RAM and PROCESSOR) will increase the speed of the process.

OPERATIONAL FEASIBILITY

Proposed system is beneficial since it turned into information system analyzing the traffic that will meet the organizations operating requirements. Internet security, the file is transferred to the destination and the acknowledgement is given to the server. Bulk of data transfer is sent without traffic.

DEVELOPMENT ENVIRONMENT

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 SYSTEM SPECIFICATION

3.1.1 SOFTWARE SPECIFICATION

The section specifies the software in which this application was developed.

Platform	:	Windows XP.
Software	:	Visual Studio 5.0(C#.Net)

3.1.2 SOFTWARE FEATURES

3.1.2.1 WINDOWS XP

Windows-XP is a powerful system. It includes many useful programs and accessories. It provides a wealth of features and it keeps the complex features relatively simply to use. It tries to create a work environment for the computers that model the actual working environment on our desk. Windows–XP has many new and enhanced features.

- **It has the facility for multitasking.**
- **It is user friendly.**
- **Every application has its own online help facility.**

3.1.2.2NET FRAMEWORK

.NET is the framework for which we develop applications. It sits in between our application programs and operating system. Applications developed for .NET run inside .NET and are controlled by .NET. It supports both Windows and web applications.

Applications developed for .NET make use of the features of .NET.

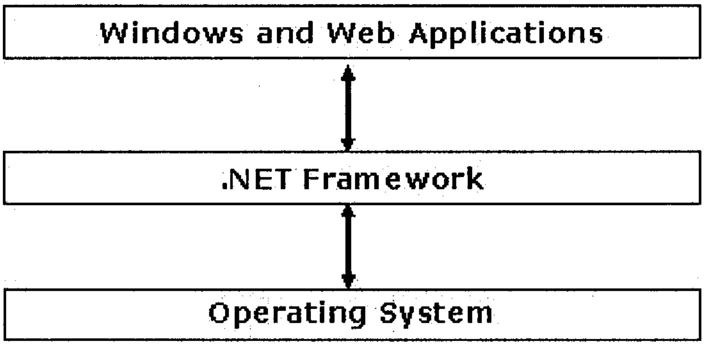


Fig 3.1 .NET Framework

.NET provides an object oriented environment. It ensures safe execution of the code by performing required runtime validations. For example, it is never possible to access an element of an array outside the boundary. Similarly, it is not possible to a program to write into another programs area, etc. The runtime validations performed by .NET makes the entire environment robust.

.NET is a "Software Platform". It is a language-neutral environment for developing rich .NET experiences and building applications that can easily and securely operate within it.

3.1.2.3 NET Framework Architecture

The .NET Framework class library facilitates types (CTS) that are common to all .NET languages. The common language runtime consists of (class loader) that load the IL code of a program into the runtime, which compiles the IL code into native code, and executes and manage the code to enforce security and type safety, and provide thread support.

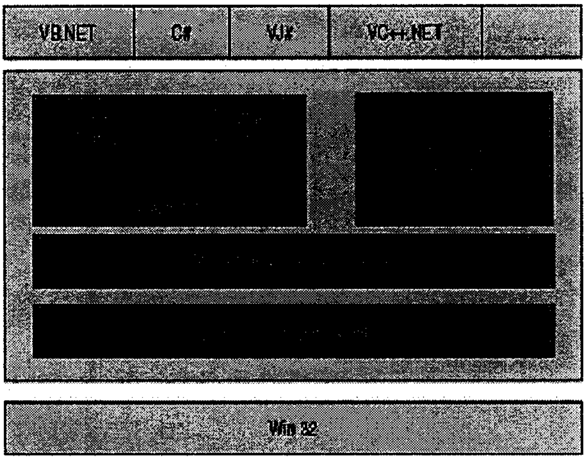


Fig 3.2 .NET Framework Architecture

.NET Framework Architecture has languages at the top such as VB .NET C#, VJ#, VC++ .NET; developers can develop (using any of above languages) applications such as Windows Forms, Web Form, Windows Services and XML Web Services. Bottom two layers consist of .NET Framework class library and Common Language Runtime. This we are going to understand using this article.

3.1.3 HARDWARE SPECIFICATION

The Section specifies the hardware components in which the application was developed and the minimum hardware configuration in which the system will work without any problem.

Processor	:	Intel Pentium III
Speed	:	1.1 GHz
RAM	:	512 MB
Hard Disk Drive	:	40 GB
Monitor	:	16-inch LG Color Monitor
Keyboard	:	104 Keys
Printer	:	Laser Printer (Canon)
Mouse	:	Logitech, 3 Mouse Button
CD-Drive	:	LG48x

SYSTEM DESIGN

CHAPTER 4

SYSTEM DESIGN

4.1 UML DIAGRAMS:

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships).

There are two types of diagrams, they are:

- STRUCTURAL DIAGRAMS
- BEHAVIORAL DIAGRAMS

4.1.1BEHAVIORAL DIAGRAMS:

The UML's five behavioral diagrams are used to visualize, specify, construct, and document the dynamic aspects of a system. The UML's behavioral diagrams are roughly organized around the major ways which can model the dynamics of a system.

4.1.1.1SEQUENCE DIAGRAM

A **sequence diagram** in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

This system contains the following objects, Such as New node, Network and Parent. Messages are passed above the arrow line. If new node is entering mean, the parent will check and return the node position. After getting that position new node is attached to the parent. Rearrange the organization and network, calculate the node address. Then update the routing table and return that address to new node.

- The following Sequence diagram describes the system action.

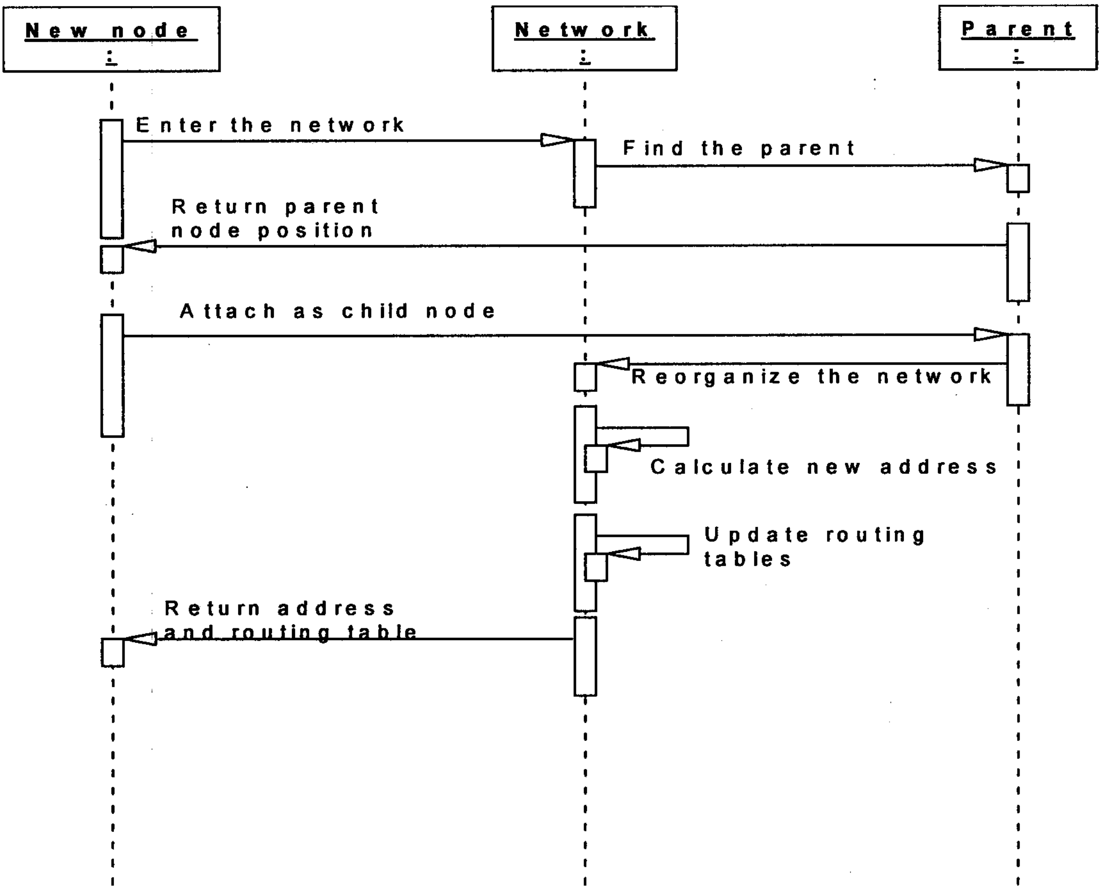


Fig 4.1 Sequence Diagram

4.1.1.2 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

In this system first state is waiting for new node attached to network. If new node is entering, Next state is computing the node position. After computation rearrange the tree and then calculate the new address. Finally update the routing table.

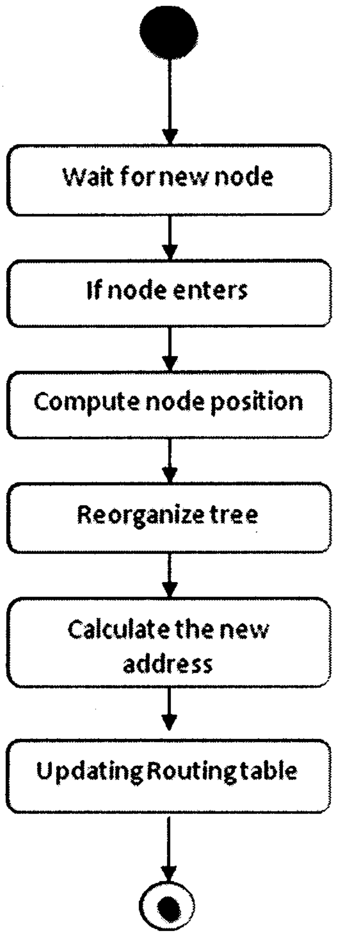


Fig 4.2 Activity Diagram

4.2 INPUT DESIGN

Input design is a process of converting user-originated inputs to a computer based format. It is an important part of development process. Since inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. It consists of developed specifications and procedures for entering data in to system, which must be in simple format for entering data easily.

Objectives of input design are:

- To avoid the errors.
- Keeping further steps simple.
- Achieving highest level of accuracy.
- Ensuring authorized access to the system.

The input files are:

- Path Establishment
- Client Creation
- File Sending
- Remove Router

4.3 OUTPUT DESIGN

Presenting the output by a computer based information system in suitable form is very important. This is accomplished with the help of output forms, which are used as a tool for taking decisions.

- Processing List
- Dynamic Routing Main

Results are generated according to the user input.

SYSTEM DEVELOPMENT

CHAPTER 5

SYSTEM DEVELOPMENT

5.1 INTRODUCTION

System development is a series of operations performed to manipulate data to produce output from computer system. This aim at translating the design of the system produced during the design phase code in user programming language. A modular approach is used for the development of the software.

When we divide a system into different modules in a logical order it will lead us into a clear solution for the problem. A module contains a group of related functions, which pass data each other. That means there is a very good correlation between the different functions in a module. Each module will accomplish a specific task or work for the project. The joint functioning of all these modules will pay way for the ultimate solution.

When we go for modularization keep the following things in mind:

- Modularization can be done in a Top-down approach or Bottom-up approach.
- The dependency between the modules should be minimum.
- The functioning of the modules must have a flow.
- The functioning inside a module must be related with each other.

The development phase for the project was created from the specifications created during the design phase. A principal activity of the development phase is coding and testing the computer program that make up the computer program component of the overall system. Other important activities include implementation, planning, equipment acquisition and system testing.

5.2 MODULE DESCRIPTION

Dynamic Routing with security consideration has four modules,

1. **Path Establishment**

2. **Client Creation**

3. **File Sending**

4. **Remove Router**

5.2.1 PATH ESTABLISHMENT

Path Establishment module deals with the information about Path between Routers. It also contains the Router Name. It is used to create the path between routers and find cost and distance.

5.2.2 CLIENT CREATION

Client Creation module deals with the information about Client. It also contains the Router Name, Source Name and Port No. It is used to create the client for specified router.

5.2.3 FILE SENDING

File Sending module deals with the information about Client file. It also contains the Router Name, File Name, and Destination.

5.2.4 REMOVE ROUTER

Remove Router module deals with the information about Removing Router. It also contains the Router Name.

SYSTEM TESTING

CHAPTER 6

SYSTEM TESTING

6.1 SYSTEM TESTING

Testing of the debugging programs is one of the most critical aspects of the computer programming triggers, without programs that works, the system would never produce the output for which it was designed. Testing is best performed when user development are asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands. **Data Processing:** The data processing tests are done to ensure that the programs are working properly and also tests whether the data are accurately updated and internal calculations are correct. **Data Output:** The test is to check whether the reports being generated are in the proper format and provide the proper information.

6.2 SYSTEM VERIFICATION

System Verification answers the question' "Am I building the product right?" It includes the review of interim work steps and interim deliverables during a project to ensure they are acceptable. Verification also determines if the system is consistent, adheres to standards, uses reliable techniques and prudent practices, and performs the selected functions in the correct manner. In file access, it verifies whether the right file is being accessed, in terms of the right place and in the right way.

For e.g., the drop downs gather from the local disk, so each dropdowns should be verified whether they are bound to the correct file. It is done during development of the key artifacts. Verification is a demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle. In Dynamic Routing with Security Consideration, Verification is done during the development itself.

6.3 SYSTEM VALIDATION

Validation answers the question “Am I building the right product?” This checks whether the developer is moving towards the right product, whether the development is moving towards the actual intended product that was agree upon in the beginning. Validation also determines if the system complies with the requirements and performs functions for which it is intended and meets the organization’s goals and user needs.

Validation is performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment. It determines the correctness of the final software product by a development project with respect to the user needs and requirements.

Validation is performed by the following module

- Path Establishment
- Client Creation
- File Sending

6.4 TYPES OF TESTING

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then the goal will be successfully achieved. A series of testing are done for the proposed system before the system is ready for the user acceptance testing.

The types are:

- Unit Testing
- Alpha & Beta Testing
- Validation Testing
- Output Testing

6.4.1 UNIT TESTING

In this testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself.

In this testing step each module is found to working satisfactorily as regard to the expected output from the module. There are some validation checks for fields also. It is very easy to find error debut in the system.

6.4.2 ALPHA & BETA TESTING

The customer conducts the alpha testing at the developer's site. The software is used in the natural setting with the developer "looking over the developer" and recording errors and usage problems. Alpha testing is conducted in controlled environment.

The beta testing is conducted at one or more customer site by the end user of the software. Unlike software testing the developer is generally not present. Therefore, beta test is live application of the software in an environment that cannot be controlled by the developer.

6.4.3 VALIDATION TESTING

In order to ensure the security of the data, the entry to the system is restricted only to the authorized users by providing user name and password. Most of the modules are designed in such a way that the user can select the input from a list of values given in the combo box. It restricts the user from giving invalid inputs. When the user leaves any Compulsory textbox empty, a warning message is displayed indicating the user to fill the text box.

Test case id	**Test case name**	**Test case desc**	**Test steps**			**Test case status**	**Test status (p/f)**
			Step	**Expected**	**Actual**		
Path Establishment	Validate Router Name	To verify that Router Name should not be same name	Select different Router name after click connect button	Alert message "Choose different Router"	Creates a path between the routers	design	**P**
Maximum Routers	Validate No. of Routers	To verify that no. of routers should not be more than four	Clicking on the screen to get the router	Alert message "Only four routers allowed"	Routers will be displayed	design	**P**
File Selection	Validate File	To verify that file selected	Clicking on browse button then choose which file want to send	Alert message "Zero index select"	Selected file is attached	design	**P**

6.4 Test Report

6.4.4 OUTPUT TESTING

After performing the validation testing, the next test is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Testing is usually done at three levels – testing the entire system - system testing, testing the modules of the system – subsystem testing, and testing the sub modules of the system – unit test.

During the system testing, the system is used experimentally to ensure that the software does not fail, i.e., it will run according to its specifications and in the way users expect. Special test data are input for processing, and the results examined. A limited number of users may be allowed surprises before the organization implements the system and depends on it.

SYSTEM IMPLEMENTATION

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 SYSTEM IMPLEMENTATION

Implementation is the final and important phase. The most critical stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective. The system can be implemented only after testing. This method offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

This involves,

- Creating computer computable files
- Training the operating staff
- Installing hardware.

This system implemented with dynamic routing algorithm. So this system becomes more user-friendly and also it will give high efficiency and low cost and destination. Following type of implementation should be carry out in this System

1. Implementation of computer system to replace a manual system. The problem encountered are converting files, training users, creating accurate files, and verifying printout for integrity
2. Implementation of a new system to replace existing one. This is usefully a difficult conversion. Some large computer systems have taken as long as year to convert.
3. Implementation of a modified application to replace an existing one, using the same computers. This type of conversion is relatively easy to handle, provide there are no major changes in the file.

SYSTEM MAINTENENCE

CHAPTER 8

SYSTEM MAINTENANCE

Software Maintenance is a set of software engineering activities that occur after software has been delivered for the customer and put into operation. It can be described as the activities that are to be undertaken after software is released for the use. The success of software and the project relies in the maintenance procedure performed.

The maintenance is performed at regular intervals to keep the project safe and reliable. Every time changes attempted on the software will cause serious and unexpected side effects. So the maintenance of the software should be considered seriously. Software maintenance is of course fair more than fixing mistakes.

Maintenance is needed in many cases such as:

- It is unreasonable to assume that software testing will uncover all latent errors in a large program.
- Rapid change is encountered in every aspect of computing.
- A successful software package may be recommended for new capabilities, further modifications of existing functions, general enhancements etc.
- The software is changed to improve future maintainability or reliability.

Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environment. It has been seen that there are always some errors found in the system that must be noted and corrected. It also means the review of the system from time to time. The review of the system is done for:

- Knowing the full capabilities of the system.
- Knowing the required changes or the additional requirements
- Solving the performance

Conclusion and Future Enhancement

CHAPTER 9

CONCLUSION AND SCOPE FOR FUTURE ENHANCEMENT

9.1 CONCLUSION

The system is developed keeping in mind that it should adapt to the future requirements to a great extend. Our project **"Dynamic Routing with Security Consideration"** is working efficiently and effectively.

The proposed algorithm should be easy to implement and compatible with popular routing protocols, such as the Routing Information Protocol (RIP) for wired networks and Destination-Sequenced Distance Vector (DSDV) protocol for wireless networks, over existing infrastructures. An analytic study was developed for the proposed algorithm and was verified against the experimental results. Our security enhanced dynamic routing could be used with cryptography-based system designs to further improve the security of data transmission over networks. The system is perfectly tested for its functionality and is appreciated for its efficiency, flexibility, and accuracy and user friendliness.

9.2 SCOPE FOR FUTURE ENHANCEMENT

Networking is a very major domain and it is keep growing. The system must support for multiple Development Environment. The system is very flexible and changes can be made without difficulty. Further extensions in the system can be made to submit more reports to the management. This will give the management a clear picture of the proceedings in the Dynamic Routing with Security Consideration. In order to modify and use the system efficiently there should be more communication between the routers.

CHAPTER 10

APPENDICES

10.1 SAMPLE CODING

1.TO FIND ALL POSSIBLEPATHS BETWEEN THE SOURCE AND THE DESTINATION

```
public void getpath(string source, string destination, ArrayList mid, string midarea)
{
arraylink = DynamicRouting.arryDistance;

for (int i = 0; i < arraylink.Count; i++)
{
rec = arraylink[i].ToString();
splt = rec.Split('_');

if (midarea.Equals(splt[1]))
{
middle = splt[2];
}
else if (midarea.Equals(splt[2]))
{
middle = splt[1];
}
else { continue; }

if (middle.Equals(destination))
{
middlearray = new ArrayList();

for (int j = 0; j < mid.Count; j++)
{
middlearray.Add(mid[j]);
}

middlearray.Add(middle);

for (int a = 0; a < middlearray.Count; a++)
{

addarea.Add(middlearray [a]);
}
```

```
for (int k = 0; k < addarea.Count; k++)
{

if ((k + 1) != addarea.Count)
{
path = addarea[k].ToString() + "_" + addarea[k + 1].ToString();

for (int x = 0; x < DynamicRouting.arryDistance.Count; x++)
{
rec = DynamicRouting.arryDistance[x].ToString();
splt = rec.Split('_');
if ((path == (splt[1] + "_" + splt[2]))||(path == (splt[2] + "_" + splt[1])))
{
km = km + Convert.ToDouble(splt[0]);

}
```

2. PATHSORT

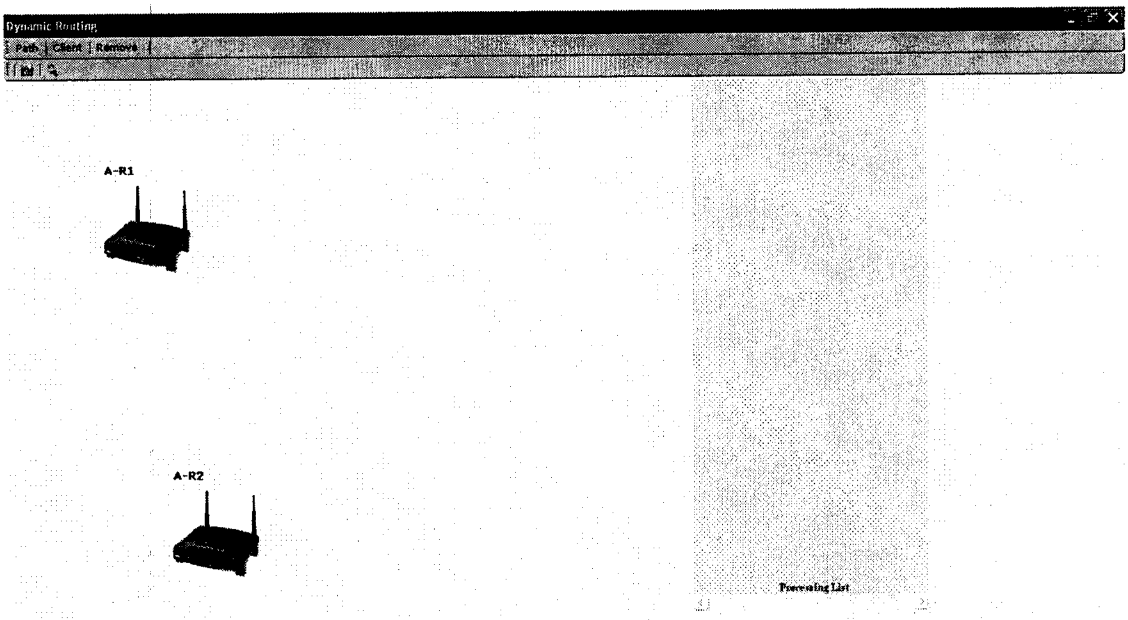
```
if (arryPathDist.Count == 3)
{
if (arryPathDist[0].ToString().Length < arryPathDist[1].ToString().Length &&
arryPathDist[0].ToString().Length < arryPathDist[2].ToString().Length)
{
Path = arryPathDist[0].ToString();
}
else if (arryPathDist[1].ToString().Length < arryPathDist[2].ToString().Length &&
arryPathDist[1].ToString().Length < arryPathDist[0].ToString().Length)
{
Path = arryPathDist[1].ToString();
}
else if (arryPathDist[2].ToString().Length < arryPathDist[0].ToString().Length &&
arryPathDist[2].ToString().Length < arryPathDist[1].ToString().Length)
{
Path = arryPathDist[2].ToString();
}
}
else if (arryPathDist.Count == 4)
{
if (arryPathDist[0].ToString().Length < arryPathDist[1].ToString().Length &&
arryPathDist[0].ToString().Length < arryPathDist[2].ToString().Length &&
arryPathDist[0].ToString().Length < arryPathDist[3].ToString().Length)
{
Path = arryPathDist[0].ToString();
}
else if (arryPathDist[1].ToString().Length < arryPathDist[0].ToString().Length &&
arryPathDist[1].ToString().Length < arryPathDist[2].ToString().Length &&
arryPathDist[1].ToString().Length < arryPathDist[3].ToString().Length)
{
Path = arryPathDist[1].ToString();
}
else if (arryPathDist[2].ToString().Length < arryPathDist[0].ToString().Length &&
arryPathDist[2].ToString().Length < arryPathDist[1].ToString().Length &&
arryPathDist[2].ToString().Length < arryPathDist[3].ToString().Length)
{
Path = arryPathDist[2].ToString();
}
else if (arryPathDist[3].ToString().Length < arryPathDist[0].ToString().Length &&
arryPathDist[3].ToString().Length < arryPathDist[1].ToString().Length &&
arryPathDist[2].ToString().Length < arryPathDist[2].ToString().Length)
{
Path = arryPathDist[2].ToString();
}
}


for (int j = 0; j < arryPathDist.Count; j++)
{

rec = arryPathDist[j].ToString();
str = rec.Split('_');
len = str.Length;


d1 = Convert.ToDouble(str[len - 1]);
if (d == d1)
{
arryPath.Clear();
for (int i = 0; i < len - 1; i++)
{
arryPath.Add(str[i]);
}

//   Path = rec;
break;
}

}
int dist = 0;
string[] paths = Path.Split('_');
```

```
if (paths.Length == 3)
{
for (int i = 0; DynamicRouting.arryDistance.Count > i; i++)
{
string[] dis = DynamicRouting.arryDistance[i].ToString().Split('_');
if ((paths[0] + "_" + paths[1]) == (dis[1] + '_' + dis[2]) || (paths[0] + "_" + paths[1]) ==
(dis[2] + '_' + dis[1]))
{
dist = dist + Convert.ToInt32(dis[0]);
}
}


string[] paths2 = Path.Split('_');
paths[paths2.Length - 1] = dist.ToString();
Path = paths[0] + "_" + paths[1] + "_" + paths[2];
}
else if (paths.Length == 4)
{
for (int i = 0; DynamicRouting.arryDistance.Count > i; i++)
{
string[] dis = DynamicRouting.arryDistance[i].ToString().Split('_');
if ((paths[0] + "_" + paths[1]) == (dis[1] + '_' + dis[2]) || ((paths[1] + "_" + paths[2]) ==
(dis[1] + '_' + dis[2])) || (paths[0] + "_" + paths[1]) == (dis[2] + '_' + dis[1]) || ((paths[1] +
"_" + paths[2]) == (dis[2] + '_' + dis[1])))
{
dist = dist + Convert.ToInt32(dis[0]);
}
}

string[] paths2 = Path.Split('_');
paths[paths2.Length - 1] = dist.ToString();
Path = paths[0] + "_" + paths[1] + "_" + paths[2] + "_" + paths[3];

}
else
{
for (int i = 0; DynamicRouting.arryDistance.Count > i; i++)
{
string[] dis = DynamicRouting.arryDistance[i].ToString().Split('_');
if ((paths[0] + "_" + paths[1]) == (dis[1] + '_' + dis[2]) || (paths[1] + "_" + paths[2]) ==
(dis[1] + '_' + dis[2]) || (paths[2] + "_" + paths[3]) == (dis[1] + '_' + dis[2]) ||(paths[0] +
"_" + paths[1]) == (dis[2] + '_' + dis[1]) || (paths[1] + "_" + paths[2]) == (dis[2] + '_' +
dis[1]) || (paths[2] + "_" + paths[3]) == (dis[2] + '_' + dis[1]))
dist = dist + Convert.ToInt32(dis[0]);
}
string[] paths2 = Path.Split('_');
paths[paths2.Length - 1] = dist.ToString();
Path = paths[0] + "_" + paths[1] + "_" + paths[2] + "_" + paths[3] + "_" + paths[4];
}
}
catch (Exception) { }
return Path;
}

public void pathsort()
{
int dist = 0;
for (int j = 0; j < arryPathDist.Count; j++)
{
string[] paths = arryPathDist[j].ToString().Split('_');
if (paths.Length == 3)
{
for (int i = 0; DynamicRouting.arryDistance.Count > i; i++)
{
string[] dis = DynamicRouting.arryDistance[i].ToString().Split('_');
if ((paths[0] + "_" + paths[1]) == (dis[1] + '_' + dis[2]))
{
dist = dist + Convert.ToInt32(dis[0]);
}
}

string[] paths2 = Path.Split('_');
paths[paths2.Length - 1] = dist.ToString();
arryPathDist[j] = paths[0] + "_" + paths[1] + "_" + paths[2];
}
else if (paths.Length == 4)
{
for (int i = 0; DynamicRouting.arryDistance.Count > i; i++)
{
string[] dis = DynamicRouting.arryDistance[i].ToString().Split('_');
if ((paths[0] + "_" + paths[1]) == (dis[1] + '_' + dis[2]) || ((paths[1] + "_" + paths[2]) ==
(dis[1] + '_' + dis[2])) || (paths[0] + "_" + paths[1]) == (dis[2] + '_' + dis[1]) || ((paths[1] +
"_" + paths[2]) == (dis[2] + '_' + dis[1])))
{
dist = dist + Convert.ToInt32(dis[0]);
}
}

string[] paths2 = Path.Split('_');
paths[paths2.Length - 1] = dist.ToString();
arryPathDist[j] = paths[0] + "_" + paths[1] + "_" + paths[2] + "_" + paths[3];

}
```

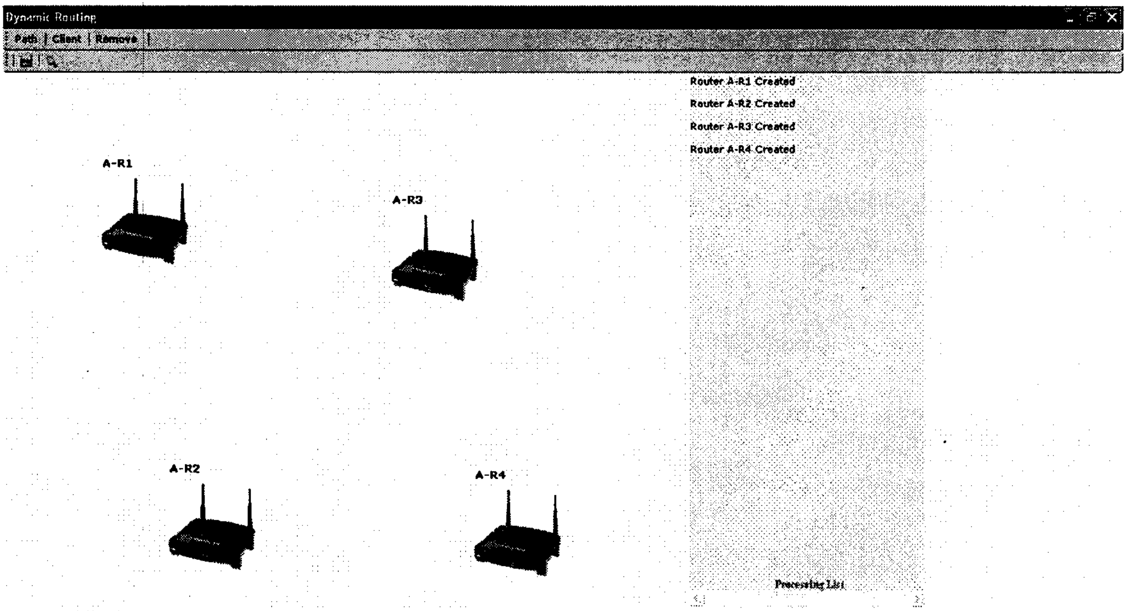
```
else
{
for (int i = 0; DynamicRouting.arryDistance.Count > i; i++)
{
string[] dis = DynamicRouting.arryDistance[i].ToString().Split('_');
if ((paths[0] + "_" + paths[1]) == (dis[1] + '_' + dis[2]) || (paths[1] + "_" + paths[2]) ==
(dis[1] + '_' + dis[2]) || (paths[2] + "_" + paths[3]) == (dis[1] + '_' + dis[2]) || (paths[0] +
"_" + paths[1]) == (dis[2] + '_' + dis[1]) || (paths[1] + "_" + paths[2]) == (dis[2] + '_' +
dis[1]) || (paths[2] + "_" + paths[3]) == (dis[2] + '_' + dis[1]))
dist = dist + Convert.ToInt32(dis[0]);
}
string[] paths2 = Path.Split('_');
paths[paths2.Length - 1] = dist.ToString();
arryPathDist[j] = paths[0] + "_" + paths[1] + "_" + paths[2] + "_" + paths[3] + "_" +
paths[4];
}
}
}
}
}
```

10.2 SCREEN SHOTS

Router Creation

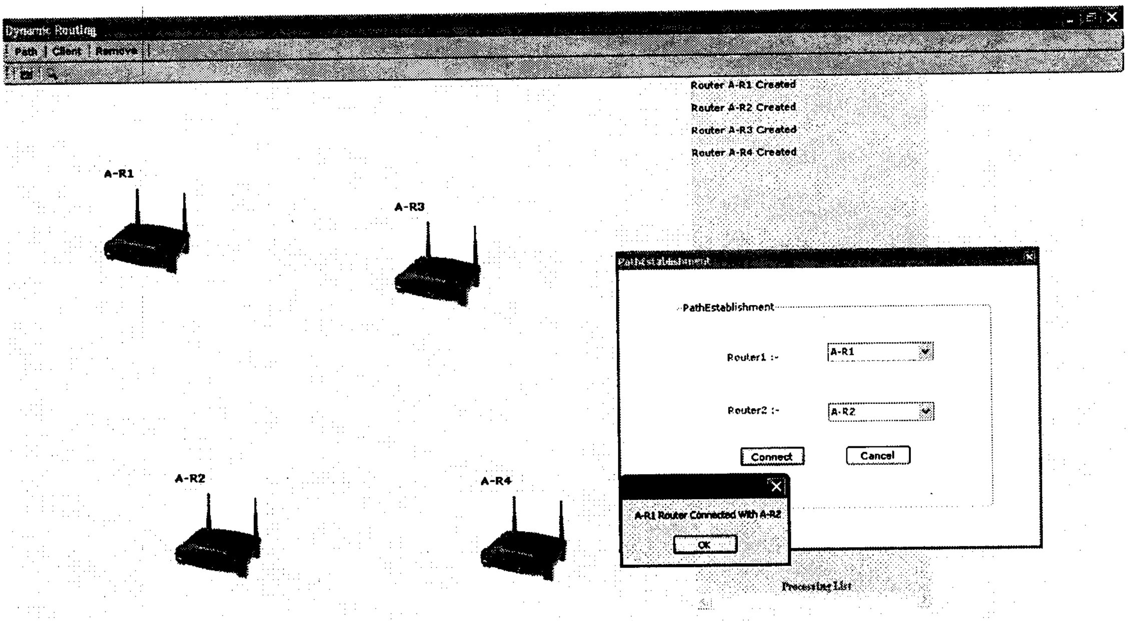


Four Routers Produced



```
Router A-R1 Created
Router A-R2 Created
Router A-R3 Created
Router A-R4 Created
```

Path Establishment



Router A-R1 Created

Router A-R2 Created

Router A-R3 Created

Router A-R4 Created

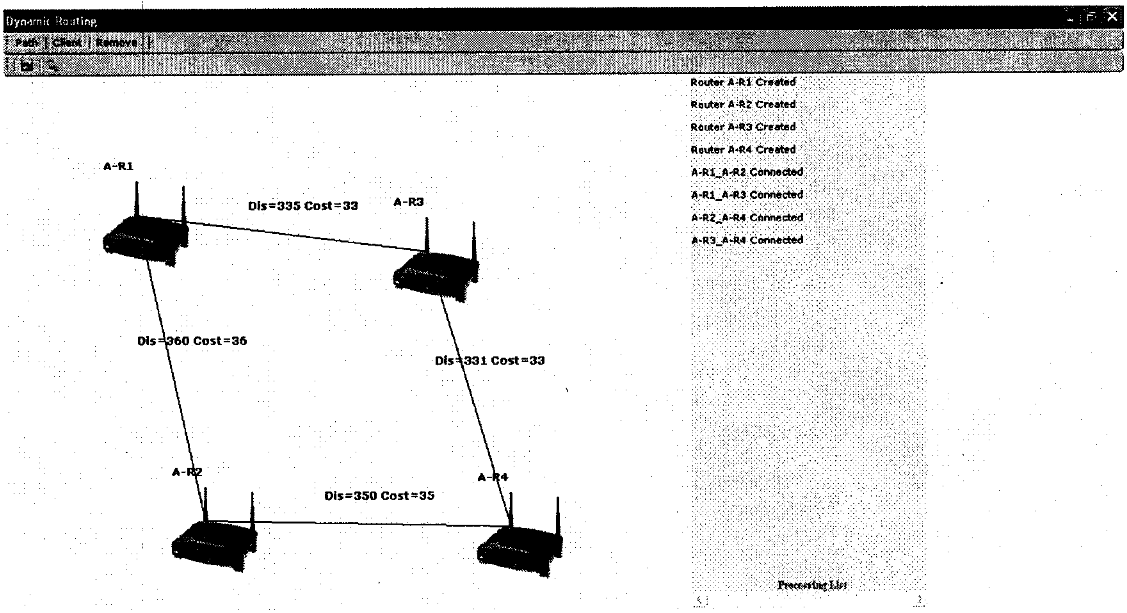
PathEstablishment

Router1 :- A-R1

Router2 :- A-R2

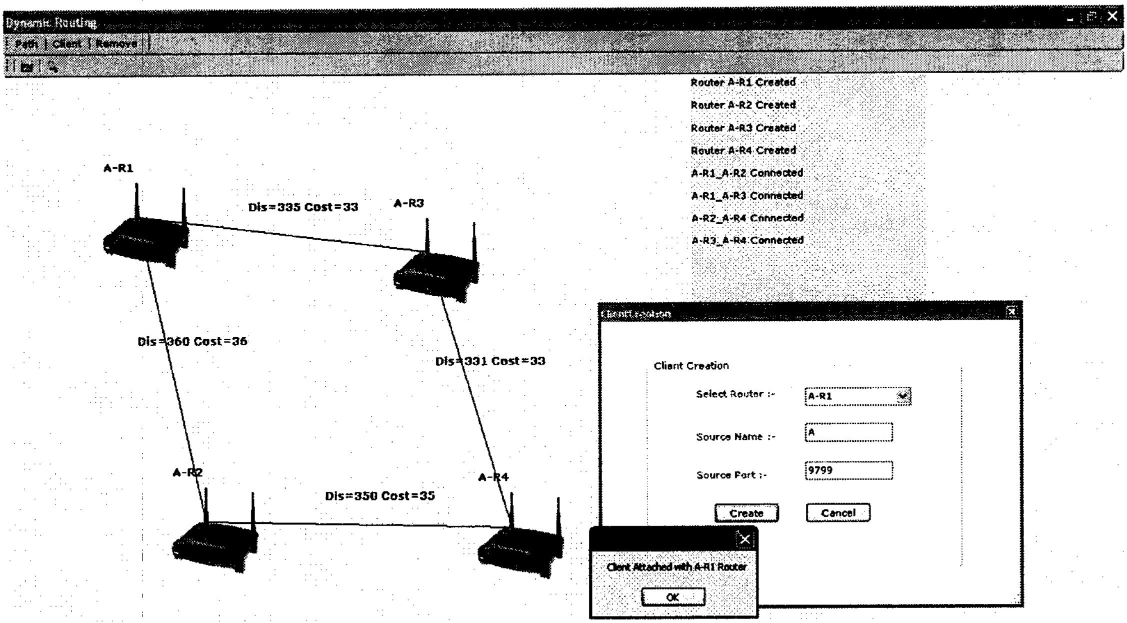
A-R1 Router Connected With A-R2

Path is established between all Routers



```
Router A-R1 Created
Router A-R2 Created
Router A-R3 Created
Router A-R4 Created
A-R1_A-R2 Connected
A-R1_A-R3 Connected
A-R2_A-R4 Connected
A-R3_A-R4 Connected
```

Client Creation



```
Router A-R1 Created
Router A-R2 Created
Router A-R3 Created
Router A-R4 Created
A-R1_A-R2 Connected
A-R1_A-R3 Connected
A-R2_A-R4 Connected
A-R3_A-R4 Connected
```

Client Creation

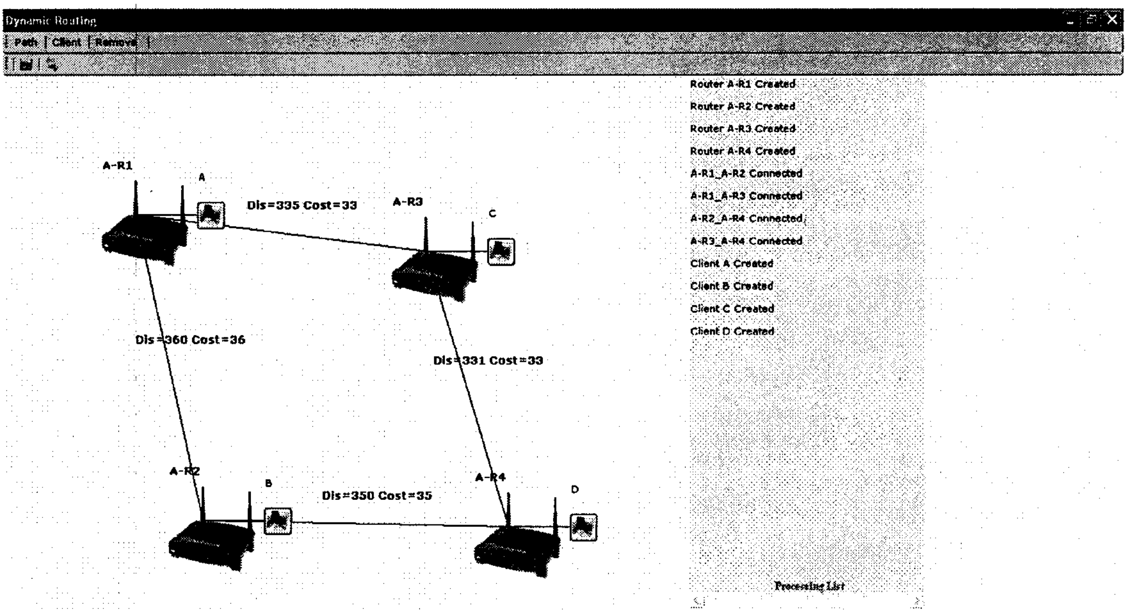
Select Router :- A-R1

Source Name :- A

Source Port :- 9799

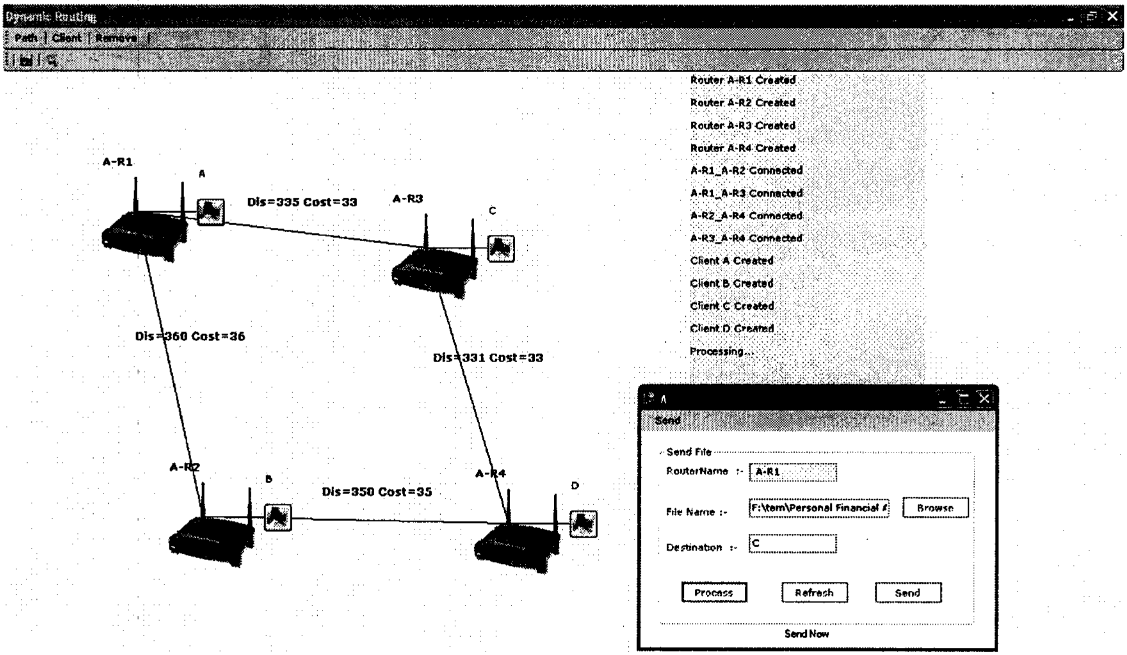
Client Attached with A-R1 Router

Creation Client for All Routers



```
Router A-R1 Created
Router A-R2 Created
Router A-R3 Created
Router A-R4 Created
A-R1_A-R2 Connected
A-R1_A-R3 Connected
A-R2_A-R4 Cnnnected
A-R3_A-R4 Connected
Client A Created
Client B Created
Client C Created
Client D Created
```

File Sending Process



```
Router A-R1 Created
Router A-R2 Created
Router A-R3 Created
Router A-R4 Created
A-R1_A-R2 Connected
A-R1_A-R3 Connected
A-R2_A-R4 Connected
A-R3_A-R4 Connected
Client A Created
Client B Created
Client C Created
Client D Created
Processing...
```

Send File

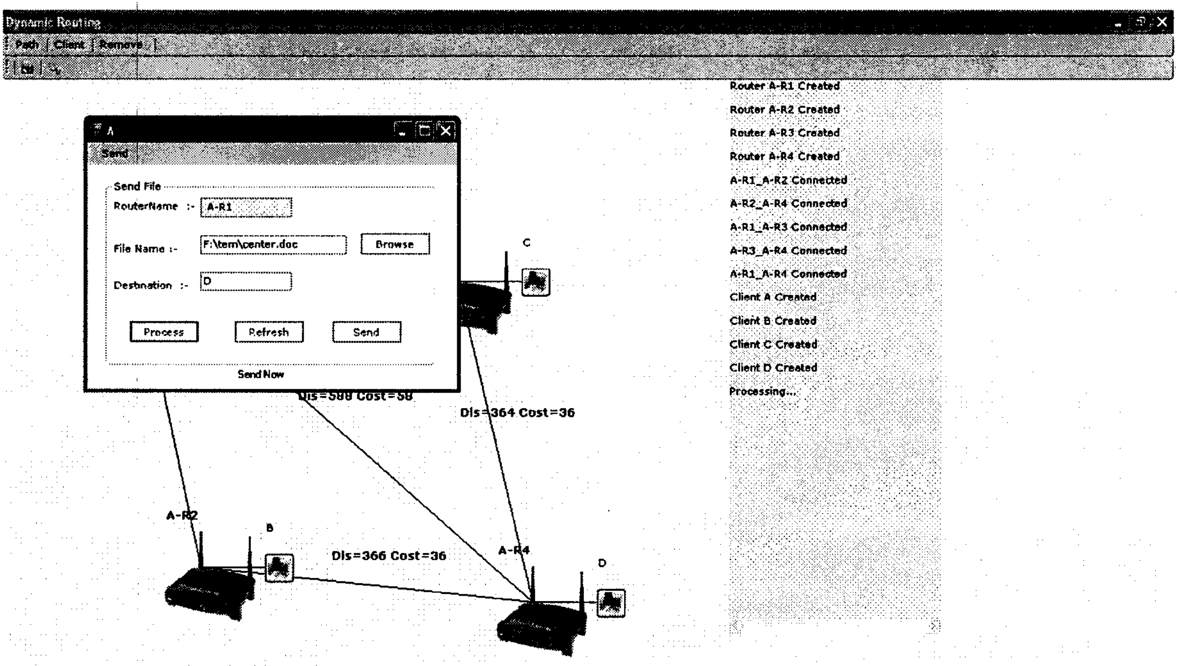
RouterName :- A-R1

File Name :- F:\tem\Personal Financial A

Destination :- C

Send Now

Client File Select Process



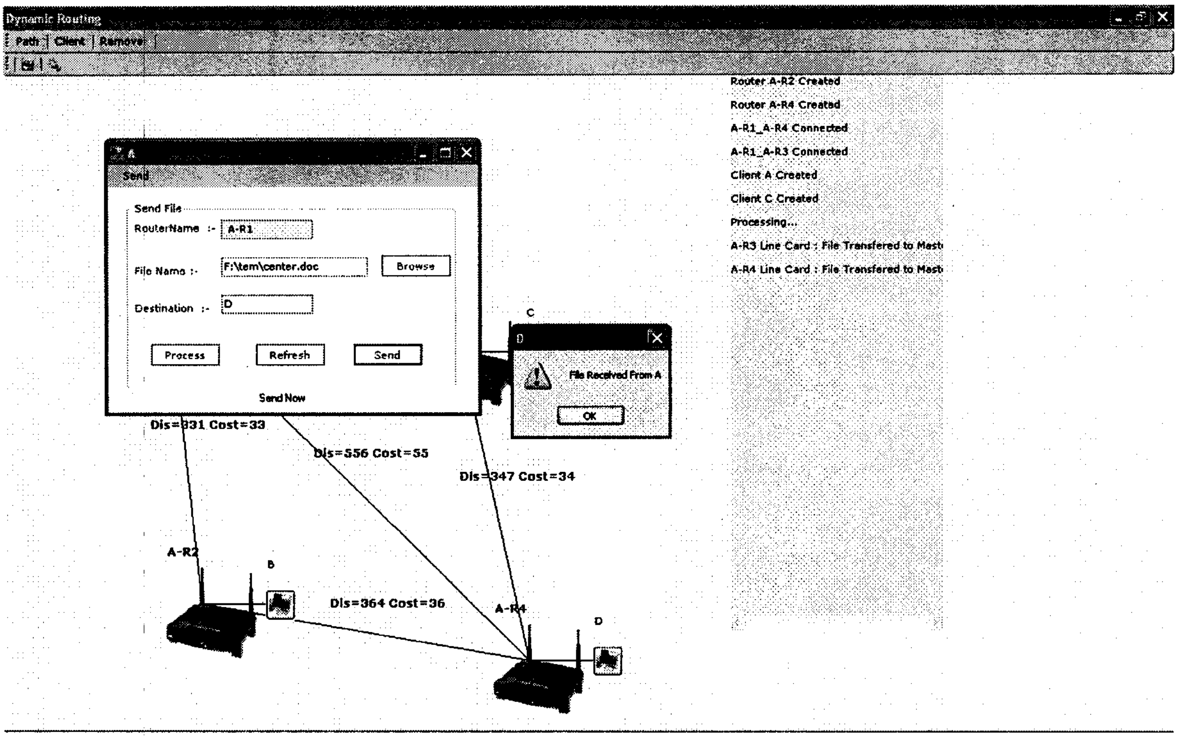
RouterName :- A-R1

File Name :- F:\tem\center.doc

Destination :- D

```
Router A-R1 Created
Router A-R2 Created
Router A-R3 Created
Router A-R4 Created
A-R1_A-R2 Connected
A-R2_A-R4 Connected
A-R1_A-R3 Connected
A-R3_A-R4 Connected
A-R1_A-R4 Connected
Client A Created
Client B Created
Client C Created
Client D Created
Processing...
```

Receiving File



RouterName :- A-R1

File Namo :- F:\tem\center.doc

Destination :- D

File Received From A

```
Router A-R2 Created
Router A-R4 Created
A-R1_A-R4 Connected
A-R1_A-R3 Connected
Client A Created
Client C Created
Processing...
A-R3 Line Card : File Transfered to Mast
A-R4 Line Card : File Transfered to Mast
```

References

CHAPTER 11

REFERENCES

IEEE

1. Manjeshwar and D. P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," inProc. Of 15th Int.Parallel and Distributed Processing Symp. Workshops, San Francisco, USA, Apr.2001.

2. Chin-Fu Kuo and Kun Chan, "Dynamic Routing with Security Consideration," Transactions on Parallel and distributed System-vol 20, Jan-2009.

BOOKS

1. Andrew S.Tanenbaum, "Computer Networks" PEARSON Prentice Hall Fourth Edition.

2. J.F. Kurose and K.W. Ross, "Computer Networking-A Top-Down Approach Featuring the Internet" Addison Wesley, 2003.

3. Cameron Wakefield, "VB.Net and the .Net Platform" Tata McGraw Hill

WEBSITES

1. www.openssl.org

2. www.isi.edu

3. www.google.com