ENSC 835: COMMUNICATION NETWORKS

MODELING AND SIMULATING SPANNING TREE PROTOCOL VS RAPID SPANNING TREE PROTOCOL ON VARIOUS NETWORK TOPOLOGIES

Date: April 17, 2011

Final Project Report – Group 6

Prepared for:
Prof. Ljiljana Trajkovic

Prepared By:
Simran Sarai (301058482, sks17@sfu.ca)
Mohammad Manjur Rashed Khan (301136637, mmkhan@sfu.ca)

Webpage:
http://www.sfu.ca/~mmkhan

Abstract

With the increasing demand of high speed Local Area Networks, the probability of attaching two layer-2 (Data Link Layer) devices together increases and this creates a switching loop. The switching loop broadcasts the frames to all the ports and since there is no Time-To-Live value attached to the Data Link layer, these frames circulate in the loop endlessly, thereby, bringing the whole network down. To avoid switching loops, IEEE 802.1d protocol defines the Spanning Tree Protocol and IEEE 802.1w enhances the Spanning Tree Protocol into Rapid Spanning Tree Protocol. Both of these protocols prevent the switching loops in the Layer-2 by creating a tree-like logical topology in the network, and cutting out the physical loop between the bridges.

In this project, we simulated a five-bridged Ring Topology using Spanning Tree Protocol (STP) and Rapid Spanning Tree Protocol (RSTP) and observed that RSTP converges a tree five times faster than a STP. Increasing the number of links did not affect the STP and RSTP performance in the network; however, increasing the number of nodes increased the end-to-end Ethernet delay between the end stations.

Acknowledgement

We would like to thank our professor Dr. Ljiljana Trajkovic for supervising this project and providing guidance and critic in each phase of this project. We would also like to thank Mr. Sorin Holban, Engineering Systems Analyst, Broadcom, for suggesting the real time networking problems associated with Data Link Layer and providing valuable suggestions from time to time.

We would also like to thank Mr. Chao for his help with technical problems with OPNET 14.0. Finally, we wish to thank our families and friends for their encouragement and support.

Table Of Contents

LIST OF FIGURES

List Of Tables

GLOSSARY

LAN	Local Area Network
STP	Spanning Tree Protocol
RSTP	Rapid Spanning Tree Protocol
IEEE	Institute of Electrical and Electronics Engineers
BPDU	Bridge Protocol Data Unit
MAC	Medium Access Control
STA	Spanning Tree Algorithm

1.0 Introduction

Many industrial applications require high availability networks and Ethernet is the main component of a network these days. Shipment of Ethernet-enabled devices on industrial networks is estimated to grow by 27.5% per year from 2008 through 2012 [1]. Ethernet enables a variety of different network topologies such as linear, star, ring, mesh etc. Cost efficient and reliable systems are high in demand and competitors adopt new approaches in creating Ethernet-enabled daisy chain topologies. These approaches create cost-effective and high performance alternative to traditional Field-bus technologies.

The main disadvantage of daisy-chained Ethernet links is the fact that the failure of one link disturbs the communication to all the devices downstream. So, ring architecture is adopted to compensate a failed link by reversing the communication path. Unfortunately, ring architecture, without specialized protocols, causes unacceptable network disturbances such as switching loops and industry standards are required to address these issues. The industry standard Spanning Tree Protocol (STP), defined by IEEE 802.1d, provides the solutions to switching loop problem. Rapid Spanning Tree Protocol (RSTP) is the enhancement of STP and is defined by IEEE 802.1w. This project discusses the performance comparisons between STP and RSTP on a ring and a mesh topology and provides configuration guidelines that enable deployment of these protocols on bridges.

1.1 Objectives and Motivation

The motivation behind this project was an incident that occurred at our workplace. One of the co-op students working in Quality Assurance team connected two switches together using a hub. The switches did not have a switching loop protection by default and thus, brought both the LANs down. This situation could have been avoided if the switches had a spanning tree protocol enabled. In this project, we model and simulate a network in OPNET14.0, by creating a switching loop and then implementing Spanning Tree Protocols on the network. The objectives of this project are:

- create a switching-loop free network using spanning tree protocols,
- analyze Spanning Tree Protocol versus Rapid Spanning Tree Protocol performance with a failure and recovery of network link on a ring topology; and

- compare Tree convergence behaviours by increasing the number of links and nodes in the network (mesh topology).

1.2 Related work

Several extensions to IEEE 802.1d and IEEE 802.1w protocols have been introduced for the Data Link Layer devices to prevent switching loops as well as enhance the tree convergence times and reduce delays. Wang et al. [2] designed a three-ring Ethernet system that controls data transmission of the three layers Frame Industrial Ethernet. They analyzed the existence of loops in the switched network, which led to the emergence of broadcast storms, resulting in the failure of exchange network. They implemented Rapid Spanning Tree Protocol and Multi Spanning Tree Protocol to create a reliable network system that could be used in modern Industry Control field.

DesRuisseaux and Electric [3] deployed Rapid Spanning Tree Protocol in ring configurations in industrial networks to meet the fault recovery timings required by a large number of Field-bus automation applications. They used mathematical formulae to calculate the recovery time depending upon the number of switches implemented in the network design. Other related works include solving wide Spanning Tree Network problems using mathematical models [4], using traffic loads to dynamically assign Spanning Tree Protocol forwarding paths [5] and simulations of networks presenting the shortest forwarding paths to improve QoS [6].

2.0 Spanning Tree Protocol

Spanning Tree Protocol (STP) is a link management protocol that is used to create logical connections between Layer-2 nodes such that any physical loop, if exists, will be broken. The first spanning tree protocol was invented in 1985 at the Digital Equipment Corporation by Radia Perlman [12]. In 1990, the IEEE published the first standard for the protocol as 802.1D, based on the algorithm designed by Perlman [13]. In an Ethernet network, multiple active paths between stations cause loops in the network. This condition confuses the switches and bridges when they see various paths to reach the stations and according to their forwarding algorithm, the switches and bridges allow duplicate frames to be forwarded. In Data Link layer, there is no Time-To-Live value attached to the frames and when the broadcasted frames are re-broadcasted, they are stuck in the loop endlessly, eventually bringing down the network. To provide path redundancy, Spanning Tree Protocol defines a tree that branches through all the bridges and switches and blocks the redundant paths into standby state. This protocol is applicable only to the bridges and the switches; and is transparent to the end stations.

2.1 Bridge Protocol Data Unit

The bridges gather information about other bridges in the network by exchanging data messages, called Bridge Protocol Data Units (BPDUs) and elect a root bridge, designated bridges and block certain links. There are two types of BPDUs – configuration BPDUs and topology change BPDUs. Configuration BPDUs are sent between bridges to establish a network topology. Topology BPDUs are sent between the bridges after a topology change has been observed in the network. Figure 1 shows the BPDU frame format and Table 1 describes the message fields.

2	1	1	1	8	4	8	2	2	2	2	2
Protocol ID	Version	Msg Type	Flags	Root ID	Root Path Cost	Bridge ID	Port ID	Msg Age	Max. Age	Hello Time	Forward Delay

Figure 1: Bridge Protocol Data Unit Format

From the BPDU, each bridge determines:

- Root Bridge: The bridge with the highest priority is the root bridge. If all the bridges have equal priority, then the bridge with the lowest MAC address is elected as root bridge.
- Designated Bridge: In each LAN, the bridge that has the least cost path to the root bridge is chosen as designated bridge. This bridge is responsible for forwarding the traffic on its LAN.
- Root Port: In each bridge, the port that has the least cost path to the root bridge is assigned as root port.
- Designated Port: The port other than the root port is chosen as designated port.

Table 1: Bridge Protocol Data Unit Configuration Message Fields and their description

Message Field	Description
Protocol Identifier	Contains the value zero
Flag	1st bit signals topology change. 2nd bit is set to acknowledge receipt of a configuration message
Root ID	Identifies the root bridge by listing its priority and ID
Root Path Cost	Contains cost of the path from bridge sending BPDU to root bridge
Bridge ID	Identifies priority and ID of the bridge sending BPDU
Port ID	Identifies port from which BPDU was sent
Message Age	Specifies amount of time elapsed since root sent BPDU on which current configuration is based
Maximum Age	Indicates when the current configuration message should be detected
Hello Time	Provides time period between root bridge configuration messages
Forward Delay	Provides length of time that bridges should wait before transitioning to a new state after topology change

2.2 Formation of Spanning Tree

Initially, all the bridges assume that they are the root bridge and send a BPDU with root ID and bridge ID as their own ID and a zero path cost as shown in Figure 2. Each bridge compares the BPDU it sent with the BPDUs it received from other bridges.

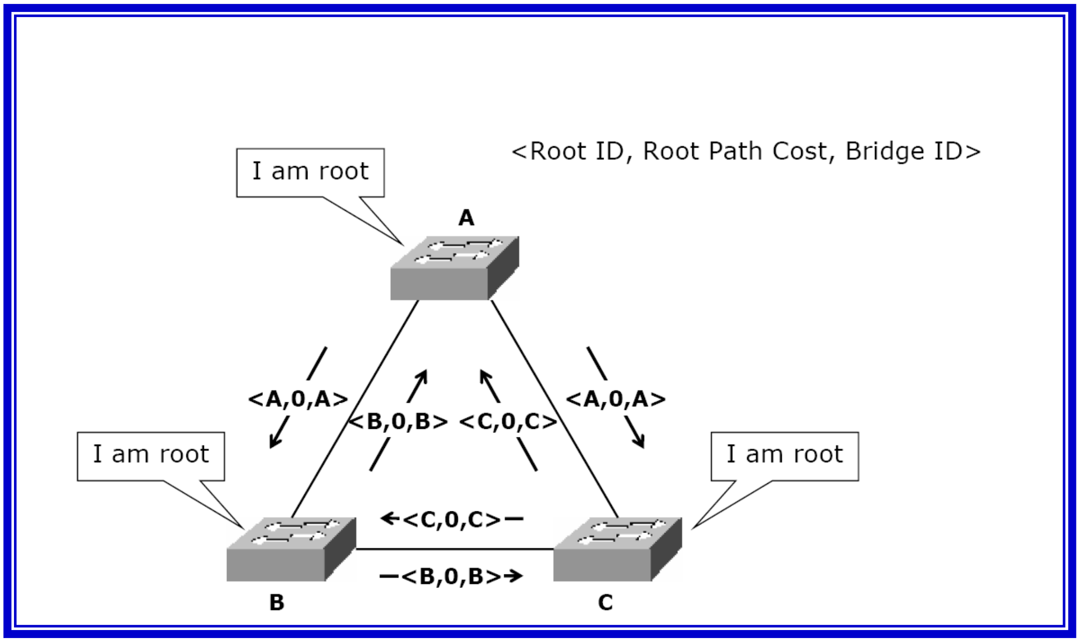

Figure 2: Initial step of Spanning Tree formation [14]

The bridge that sent a BPDU with the lowest bridge ID is chosen as the root bridge. In Figure 3, Bridge A has the smallest MAC ID and is chosen as the root bridge. The next BPDU broadcast by bridge A has: A as its root ID, 0 as its path cost and A as its bridge ID. The next BPDU broadcast by Bridge B has: A as its root ID, 1 as its path cost and B as its bridge ID. The next BPDU broadcast by Bridge C has: A as its root ID, 1 as its path cost and C as its bridge ID.

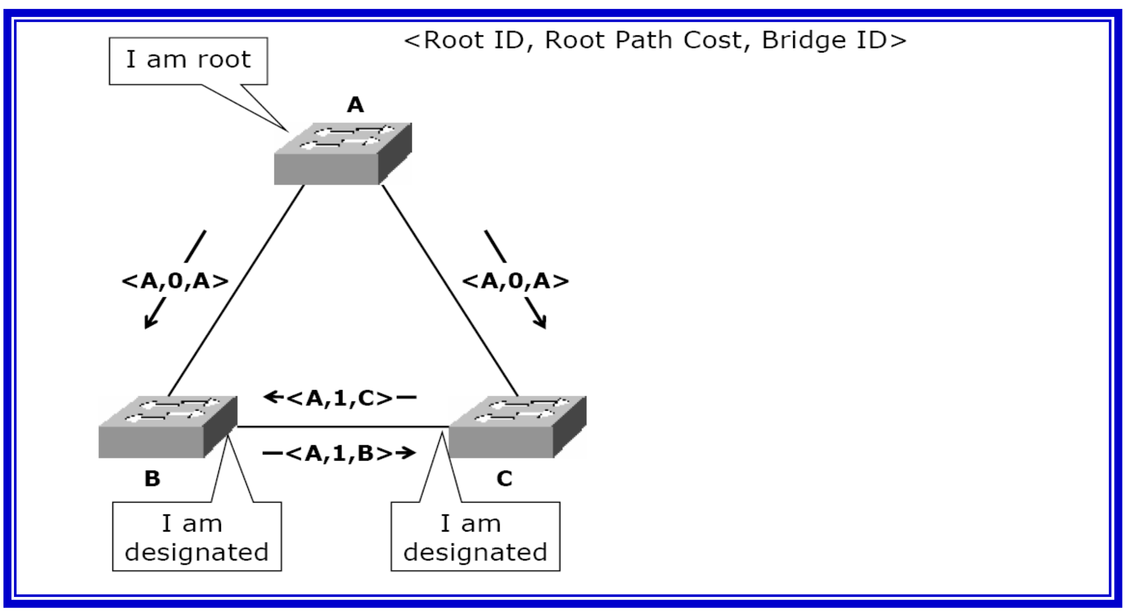

Figure 3: Determine Root and Designated switches in Spanning Tree formation [14]

Based on the BPDUs sent and received by Bridge B and Bridge C, they determine the designated bridge. Since the least path cost to the root bridge is same from both the Bridges B and C, the designated bridge is selected based on the lowest bridge ID. In this case, Bridge B's ID is lower than Bridge C's ID and therefore, Bridge B is chosen as the designated bridge on LAN 2. The port of Bridge C that connects to LAN2 is blocked and therefore, the switching loop is broken as shown in Figure 4.

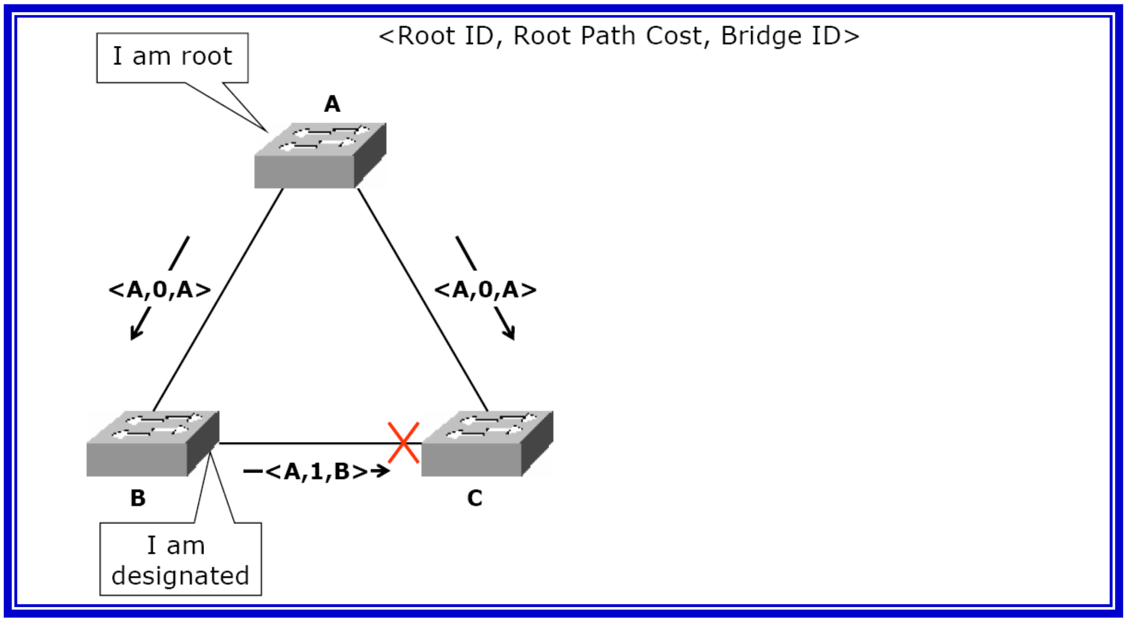

Figure 4: Spanning Tree formed with disabled links [14]

2.3 Spanning Tree Protocol Port States

Each port in a bridge that uses Spanning Tree Protocol exists in one of the five states – Blocking, Listening, Learning, Forwarding and Disabled. The flowchart in Figure 5 shows the transitioning of states in a port. When a spanning tree is configured, each port goes through blocking state and the transitory states of listening and learning at power up and then stabilizes to either the forwarding state or blocking state. During the convergence process, no data communication passes through the bridges involved.

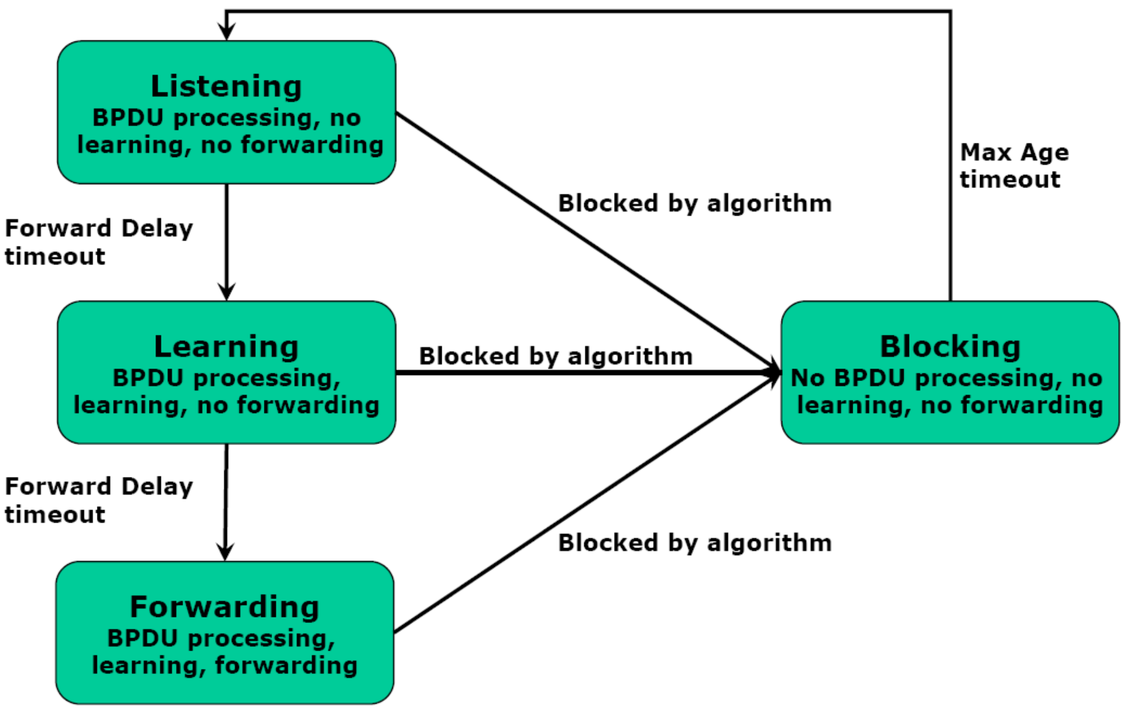

Figure 5: Flowchart representing port states in Spanning Tree Protocol [14]

2.3.1. Blocking State

After initialization, blocking state is the first state of a port. A port can also transition to the blocking state if the bridge receives a topology change notification BPDU. Whenever the network topology changes, STP blocks all the ports until the STP convergence process is started. In the blocking state, a port performs as follows:

- Discards frames received from LAN and other port of the bridge
- Receives and analyzes BPDUs and forwards them to LAN
- Does not transmit BPDUs generated by its Bridge

When a link fails, the bridges connected to the failing link wait for maximum age time given in the BPDU before they start the STP convergence process.

2.3.2. Listening State

After initialization, listening state is the second state in which the port transitions. In this state, bridges communicate using BPDUs to assign a port type to the listening port. In the listening state, a port performs as follows:

- Discards frames received from LAN and other ports in the bridge
- Receives and sends BPDUs without populating MAC address table of the bridge
- Processes BPDUs generated by its Bridge

2.3.3. Learning State

In the learning state, a port prepares to participate in frame forwarding. In the learning state, the bridge gathers information about the MAC address reachable on each port. In learning state, a port performs as follows:

- Discards frames from the LAN and other ports of its bridge
- Receives and sends BPDUs with MAC address table of the bridge
- Processes and transmits BPDUs generated by its Bridge

The listening and learning time make up the forwarding delay time in a BPDU, which is 15 seconds by default.

2.3.4. Forwarding State

In the forwarding state, a port is assigned as a root port, designated port or a blocking port. If the port is either a root or a designated port, then it forwards the frames and performs as follows:

- Forwards frames from the LAN and other ports of its bridge
- Receives BPDUs and forwards them to LAN
- Processes and transmits BPDUs generated by its Bridge

2.3.5. Disabled State

In the disabled state, a port does not participate in frame forwarding and other operations of spanning tree. A disabled port performs as follows:

- Discards frames from the LAN and other ports of its bridge
- Discards received BPDUs and does not forward them to LAN
- Does not receive BPDUs generated by its Bridge

3.0. Rapid Spanning Tree Protocol

Rapid Spanning Tree Protocol is the enhancement of Spanning Tree Protocol and is standardized as IEEE 802.1w. The need for this refined protocol arose when the recovery of connectivity provided by STP, after an outage, was deemed too slow. RSTP is backward compatible with STP and bridges using RSTP can interoperate with legacy bridges using STP. Most of the parameters used in IEEE 802.1d were left unchanged in IEEE 802.1w and so, only the differences between the two protocols will be introduced in this section.

3.1 RSTP Port States

Rapid Spanning Tree Protocol has three states – learning, forwarding and discarding. In the Spanning Tree Protocol, there was no difference between the blocking state and listening state, from an operational point of view. Both the states discarded frames and MAC IDs of other bridges. In listening state, a port can be either a designated port or a root port since it is in the transition to the forwarding state. Once in forwarding state, there is no way to determine from the port state whether the port is root or designated. This shows the failure of state-based terminology used by STP. RSTP decouples the role and the state of a port to address this issue, which is discussed in Section 3.2. A comparison between the STP port states and RSTP port states in given in Table 2.

Table 2: Port State differences between STP and RSTP [15]

STP (802.1D) Port State	RSTP (802.1w) Port State	Is Port included in Current Tree Topology?	Is Port learning MAC Addresses?
Disabled	Discarding	No	No
Blocking	Discarding	No	No
Listening	Discarding	Yes	No
Learning	Learning	Yes	Yes
Forwarding	Forwarding	Yes	Yes

3.2. RSTP Port Roles

In Spanning Tree Protocol, a port could have three roles – Root, Designated and Blocked. In Rapid Spanning Tree Protocol, the blocked role is split into the backup and alternate port roles. These ports reduce STP convergence time by allowing bridges in the LAN to have failover plans in the event that their root port or designated port fails. The Spanning Tree Algorithm (STA) determines the role of each port based on BPDUs. The port that has the least path cost to the root bridge is the root port. On each LAN, the port of a bridge that has least path cost to the root bridge is given the designated port role. In case of root port failure, an alternate port allows the bridge to use it as root failover port without going through STP convergence process. Similarly, in case of designated port failure, a backup port allows the bridge to use it as designated failover port without going through STP convergence process.

4.0 Simulations

We used OPNET 14.0 to model and simulate a Local Area Network with Ring and Mesh Topologies. Each simulation was run for 300 seconds and had four categories:

- Ring Topology scenarios with STP versus RSTP tree convergence comparison
- Ring Topology with Failure/Recovery scenarios with STP versus RSTP tree reformation comparison
- Ring Topology scenario versus scenario with increased number of links (Mesh Topology)
- Ring Topology scenario versus scenario with increased number of nodes (Mesh Topology)

4.1 Ring Topology

The first part of the simulation was to create a Ring Topology using five bridges and three workstations as shown in Figure 6. One of the workstations was configured as a source and the other two were configured as sinks. This ring topology created a physical loop which could cause switching loop problems as discussed in Section 2.0.

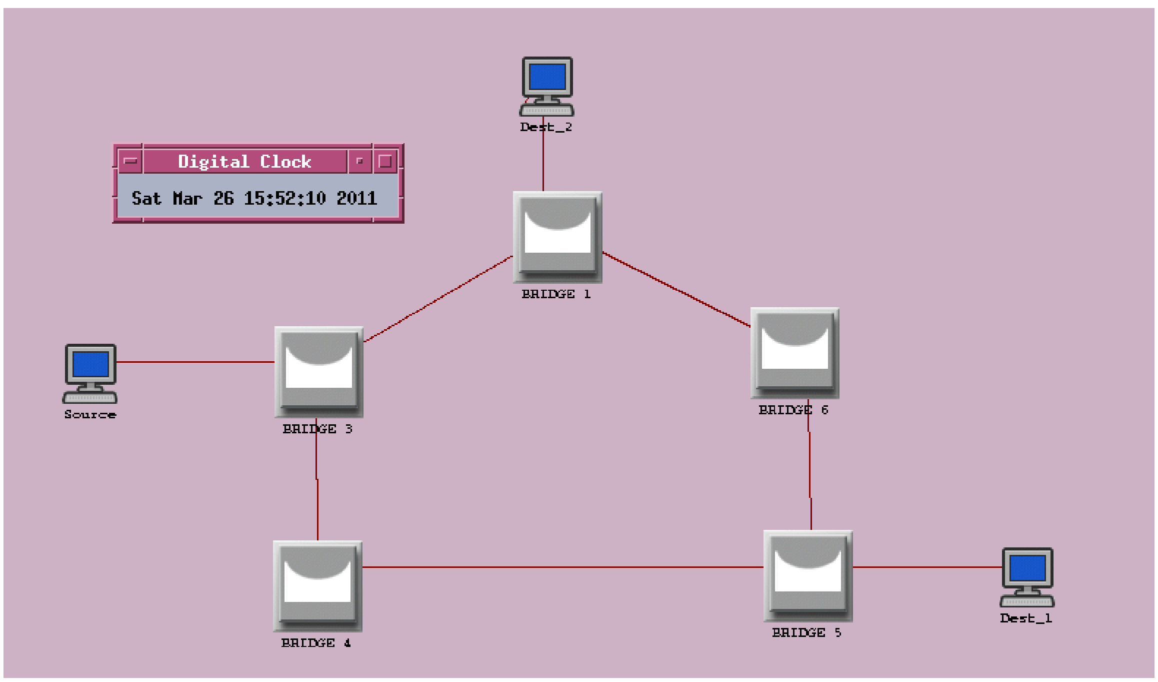

Figure 6: Ring Topology with five bridges

When the simulation started, the bridges sent BPDU frames to each other and elected a Root Bridge based on the highest priority and lowest MAC ID criteria. In this simulation, all the bridges had equal priorities, so the selection of root bridge was based on the MAC IDs of the bridges which are given in Table 3.

Table 3: Bridges and their MAC IDs for determining the root bridge

Switch	MAC ID
Switch A	0.0.A
Switch B	0.0.29
Switch C	0.0.4
Switch D	0.0.8
Root	0.0.0

After the simulation completed, spanning tree was formed which is shown in Figure 7. As seen in the Figure 7, Bridge 4 was elected as the root bridge.

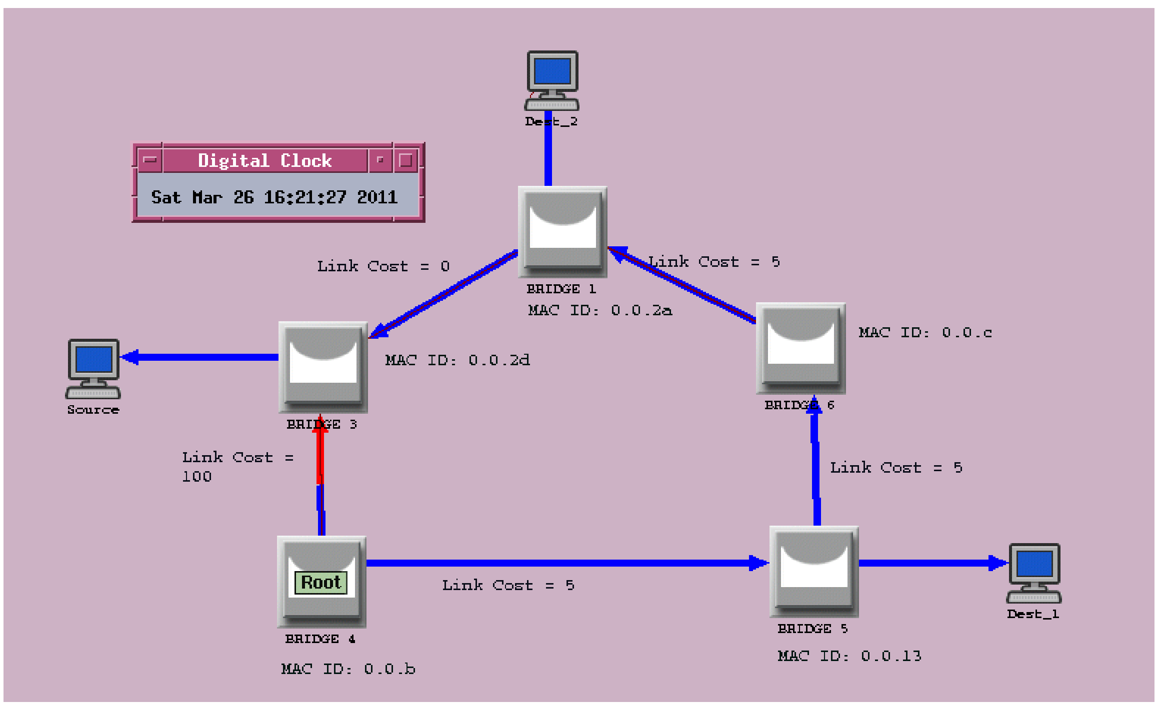

Figure 7: Spanning Tree Virtual Connections on Ring Topology

4.1.1 Results and Analysis

From the simulations, we observed that the Spanning Tree Protocol took 30 seconds to form a tree whereas Rapid Spanning Tree Protocol took 6 seconds. Figure 8 shows the point-to-point Ethernet delay that focuses on the initial convergence time. The red line represents the Ethernet delay in the network using STP and the blue line represents the Ethernet delay in the network using RSTP.

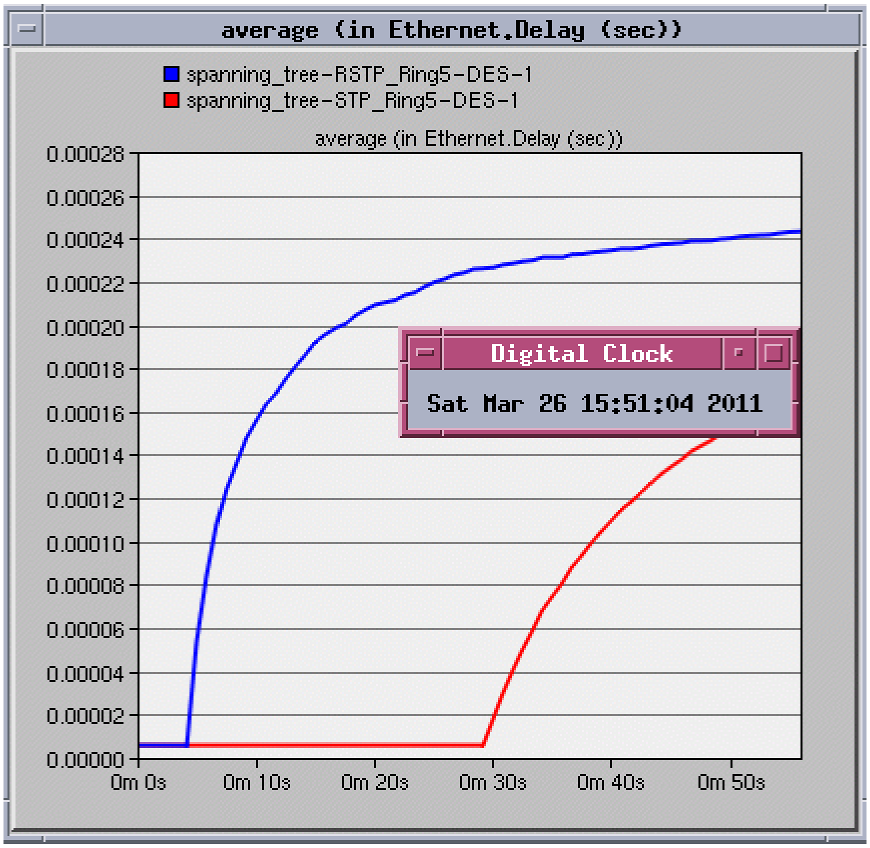

Figure 8: Ethernet delay comparison between STP and RSTP on Ring Topology

The reason for fast tree formation by RSTP over STP is due to the high BPDU traffic sent by the RSTP configured bridges initially. As seen in Figure 9, RSTP sends more BPDU traffic than STP, while forming the spanning tree. In RSTP, the BPDU traffic in the network reduces from 10,000 bits/sec through to 1000 bits/sec. In STP, the BPDU traffic in the network reduces from 3,500 bits/sec though to 1000 bits/sec.

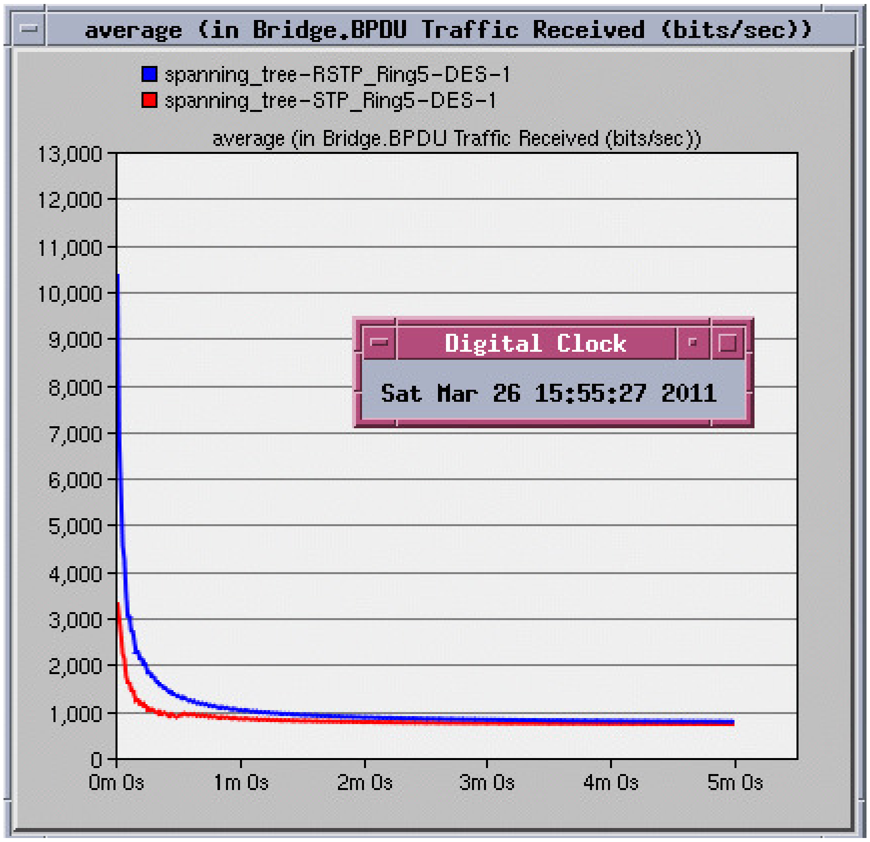

Figure 9: BPDU frames received in RSTP versus STP

4.2 Ring Topology with Failure/Recovery

In this part, we added a Failure/Recovery node model to the Ring Topology. During the simulation, Failure/Recovery node model breaks a link at 2 minutes and recovers it back at 3 minutes. The spanning tree algorithm will detect the link failure and start recalculating the spanning tree. More BPDU frames are sent in the reformation of spanning tree and when the new tree is formed, the link that was previously blocked, will be activated to forward the traffic.

The attributes of failure recover node model were configured as shown in Figure 10. The link between Bridge 4 and Bridge 5 is failed at 120 seconds and is recovered back at 180 seconds.

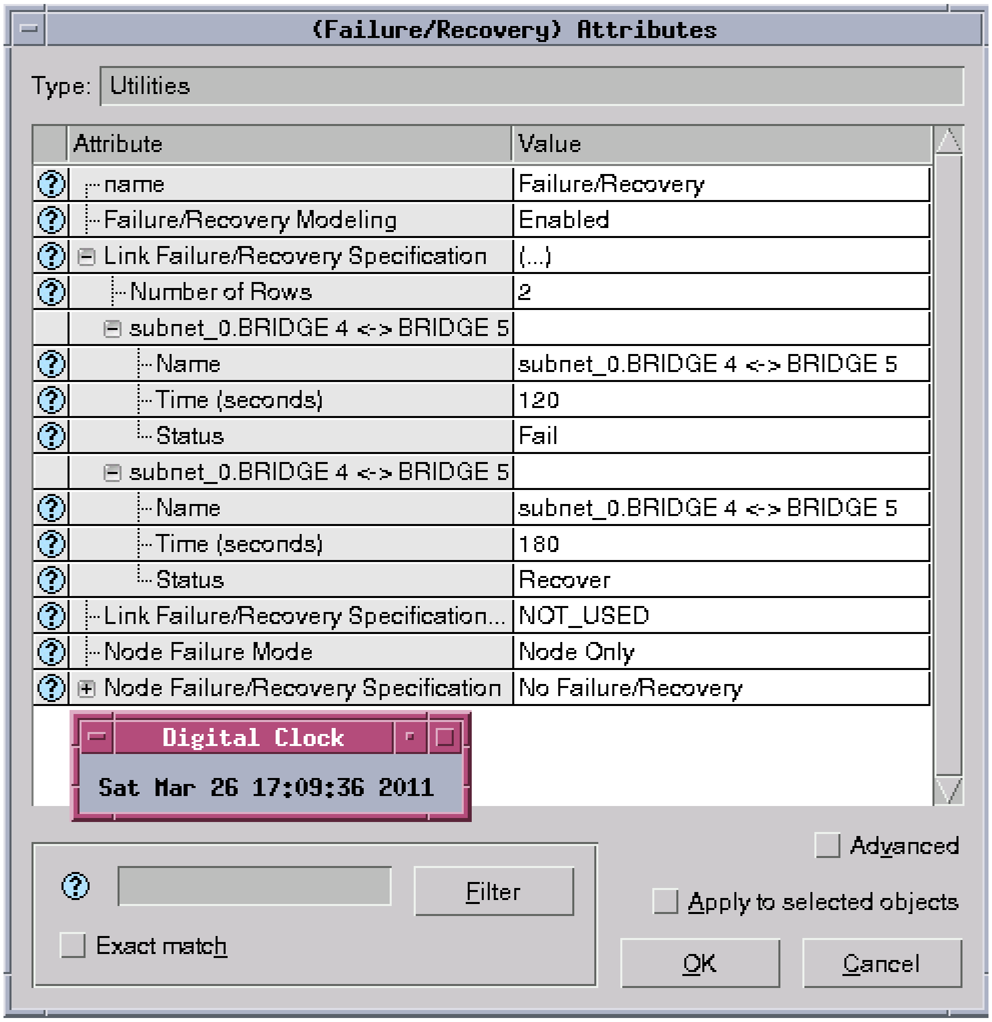

Figure 10: Failure/Recovery Node Model Attributes

4.2.1 Results and Analysis

Before the Failure/Recovery node was added to the scenario, data was collected for comparison purposes. As seen in Figure 11, the link Bridge 3 <-> Bridge 4 was blocked during the initial spanning tree formation and the link Bridge 4 <-> Bridge 5 had continuous throughput. The throughput of the blocked link is almost zero and the throughput of active link is 0.5 packets/sec.

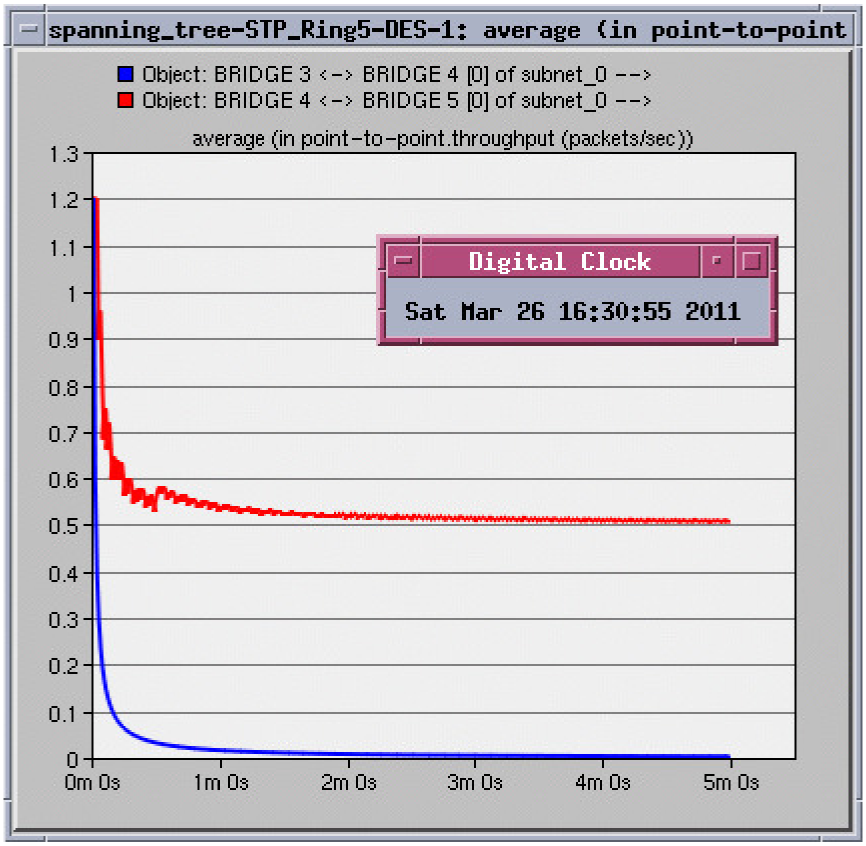

Figure11: Throughput of blocked link vs active link using spanning tree

When the Failure/Recovery node model is applied to the scenario, the bridges detect a link failure at 2 minutes and start forming a spanning tree by sending BPDU frames. The top graph in Figure 12 shows the link that was initially blocked and the bottom graph shows the link that breaks down at 2 minutes and recovers back at 3 minutes. As seen from the Figure 12, the throughput of the link Bridge 4 <-> Bridge 5 drops to zero at 2 minutes and recovers back at 3 minutes. After the link fails, it takes 44 seconds to reform the new tree and the link that was initially blocked by spanning tree, Bridge 3 <-> Bridge 4, activates.

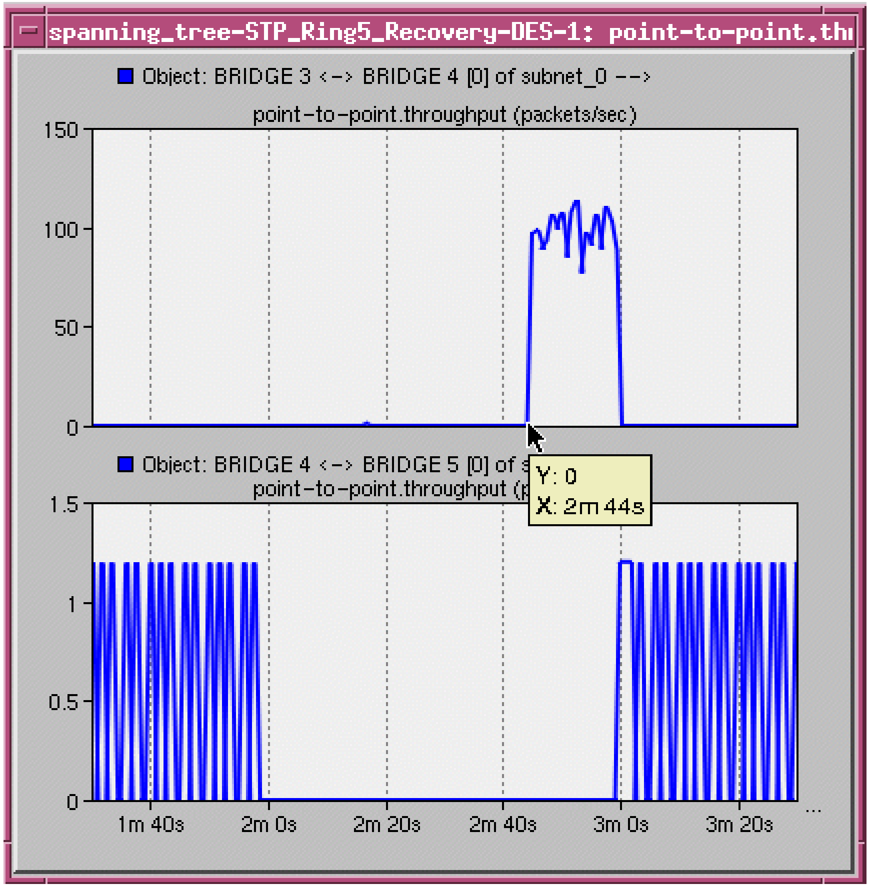

Figure 12: Throughput of failed link versus throughput of blocked link of STP

Similar analysis was done using RSTP. As seen in Figure 13, RSTP reforms the tree in 10 seconds and activates Bridge 3 <-> Bridge 4. This comparison shows that RSTP reforms a spanning tree faster than STP.

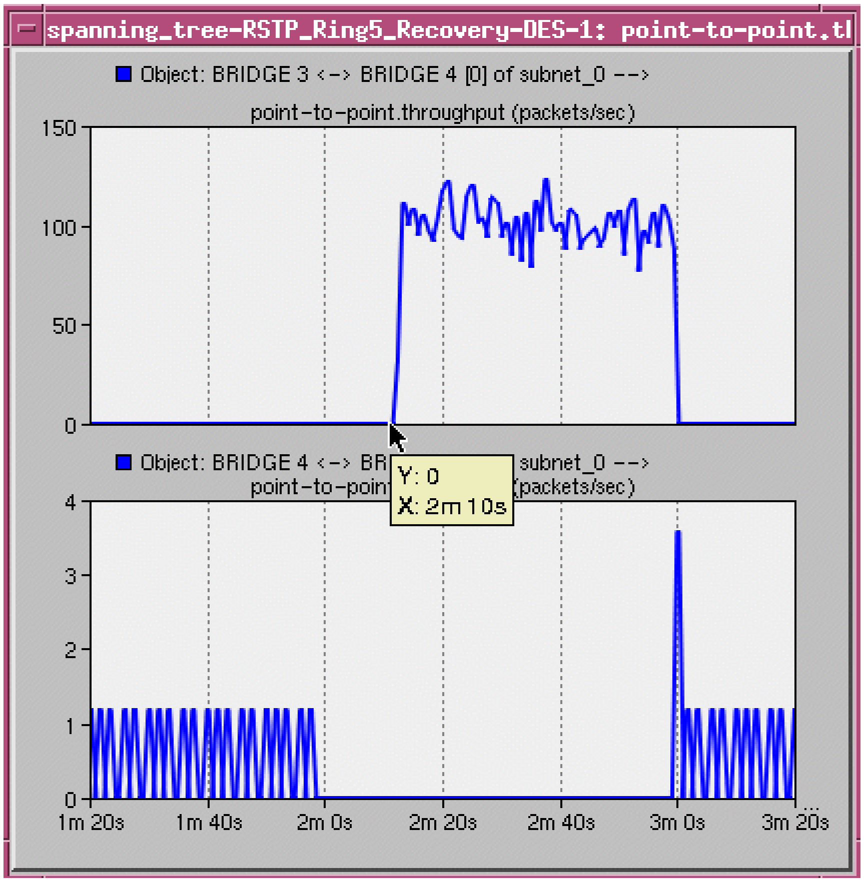

Figure 13: Throughput of failed link vs throughput of blocked link using RSTP

The BPDU frames are exchanged between the bridges to detect switching loops in a network topology. A comparison of BPDU traffic usage on Ring topologies without failure/recovery and with failure/recovery is shown in Figure 14. The top graph shows a steady BDPU traffic usage after the spanning tree is formed and the bottom graph shows a spike in BPDU traffic at 2 minutes and 3 minutes. This spike represents the high BPDU traffic exchanged between the bridges to reform a spanning tree due to the failure and recovery of one of the links.

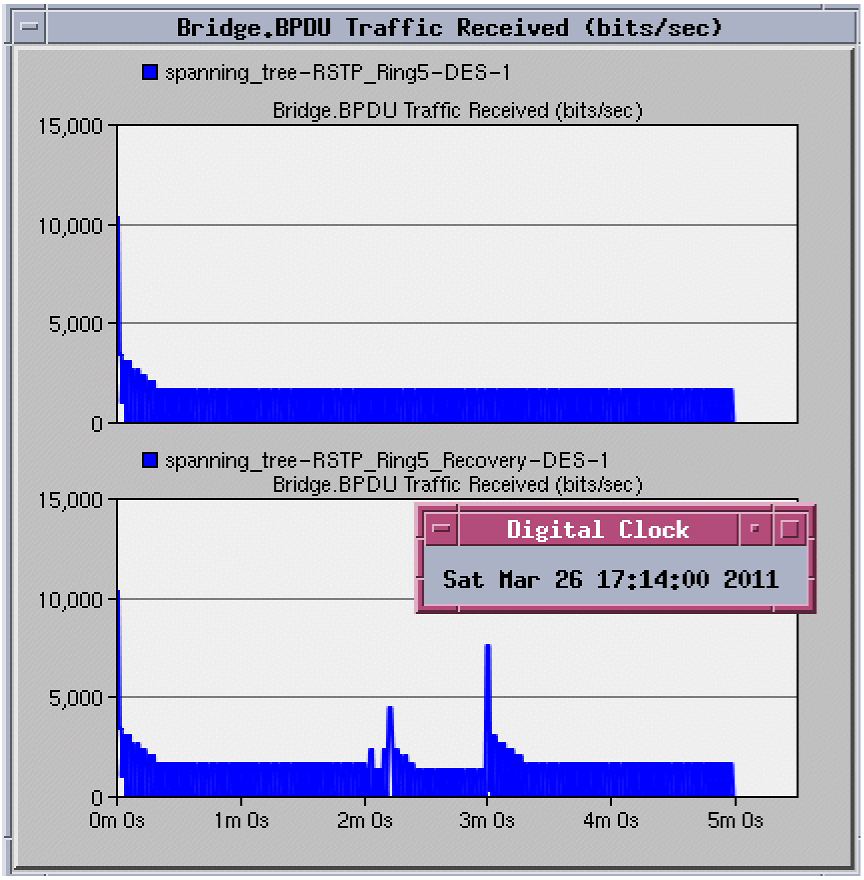

Figure 14: BPDUs received with and without link failure

4.3 STP vs RSTP in Mesh Topology

The Mesh Topology used in this scenario was based on the Ring Topology and added more links between the bridges as shown in Figure 15. This provided us the grounds to compare the spanning tree performance when the number of links are increased in a Ring Topology and the number of nodes are kept constant.

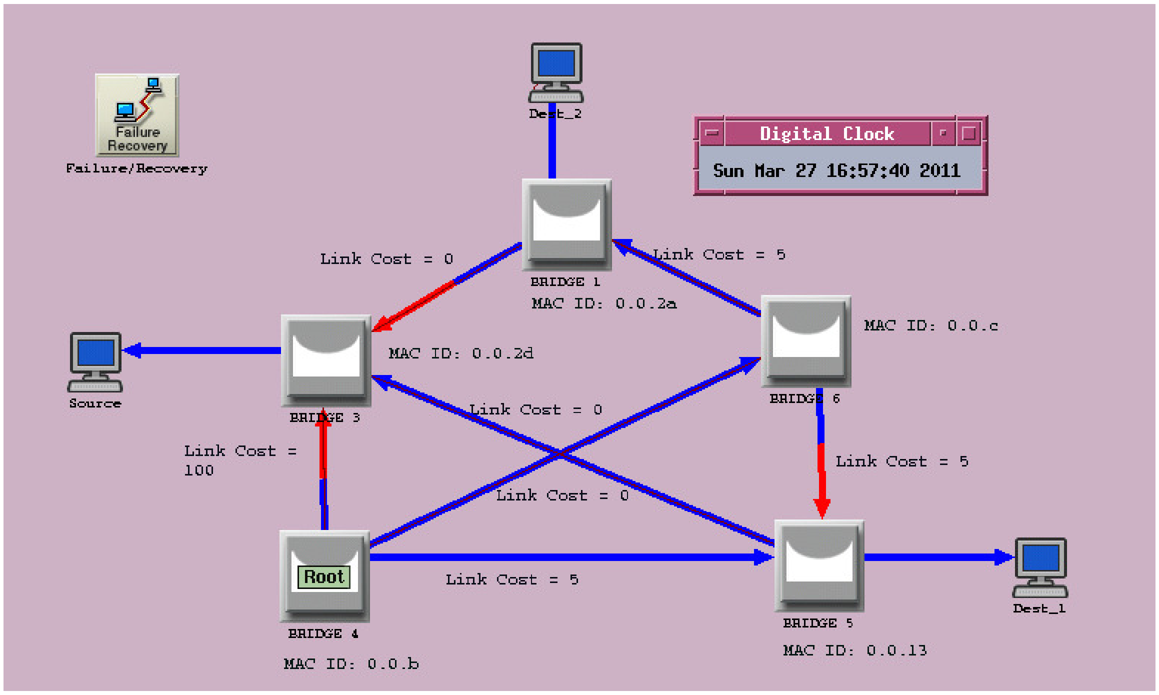

Figure 15: Ring Topology with increased number of links to form a Mesh Topology

4.3.1 Results and Analysis

An Ethernet delay comparison was made between the Ring topology and the Mesh topology. Figure 16 shows Ethernet delay using RSTP between Ring and Mesh Topology. The total path cost in Ring topology is smaller than the total path cost in Mesh topology. Therefore, the Ethernet delay is smaller in Ring topology than in Mesh topology. The blue line in Figure 16 shows the Ethernet delay in Mesh network and it reaches to 0.28 ms; and the red line shows the Ethernet delay in Ring network and it reaches to 0.25 ms.

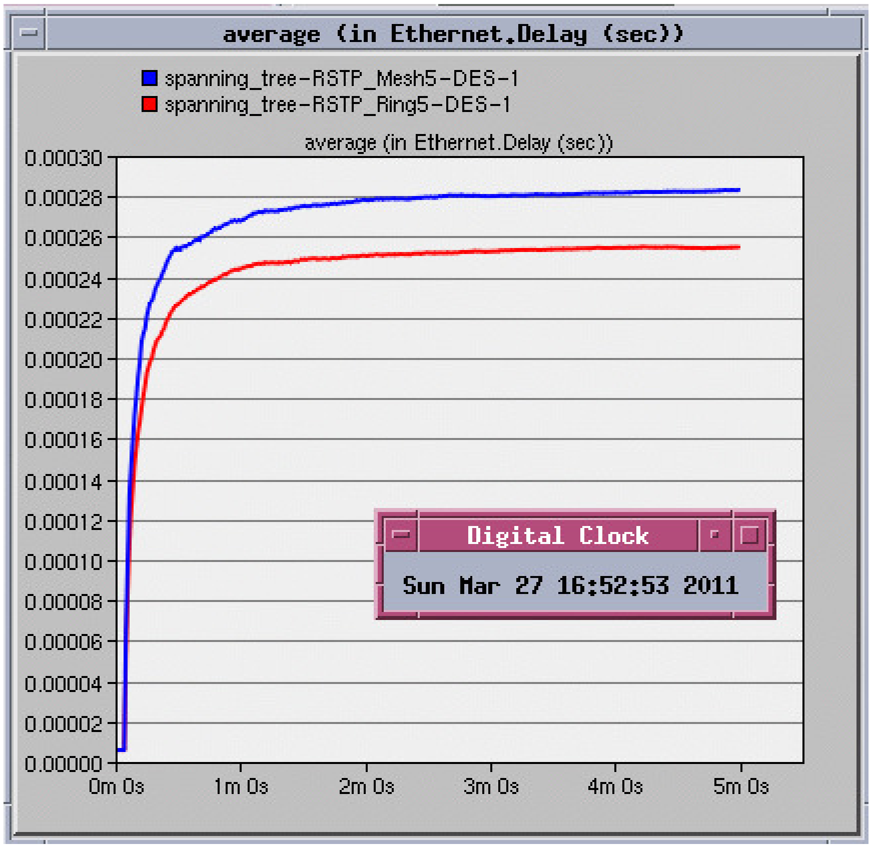

Figure 16: Ethernet delay comparison between Ring and Mesh topology using RSTP

Similar performance comparisons were made between Ring Topology and Mesh Topology using STP, which is shown in Figure 17. We observed the same trends in STP and the Ethernet delay was lower in Ring topology than in Mesh topology.

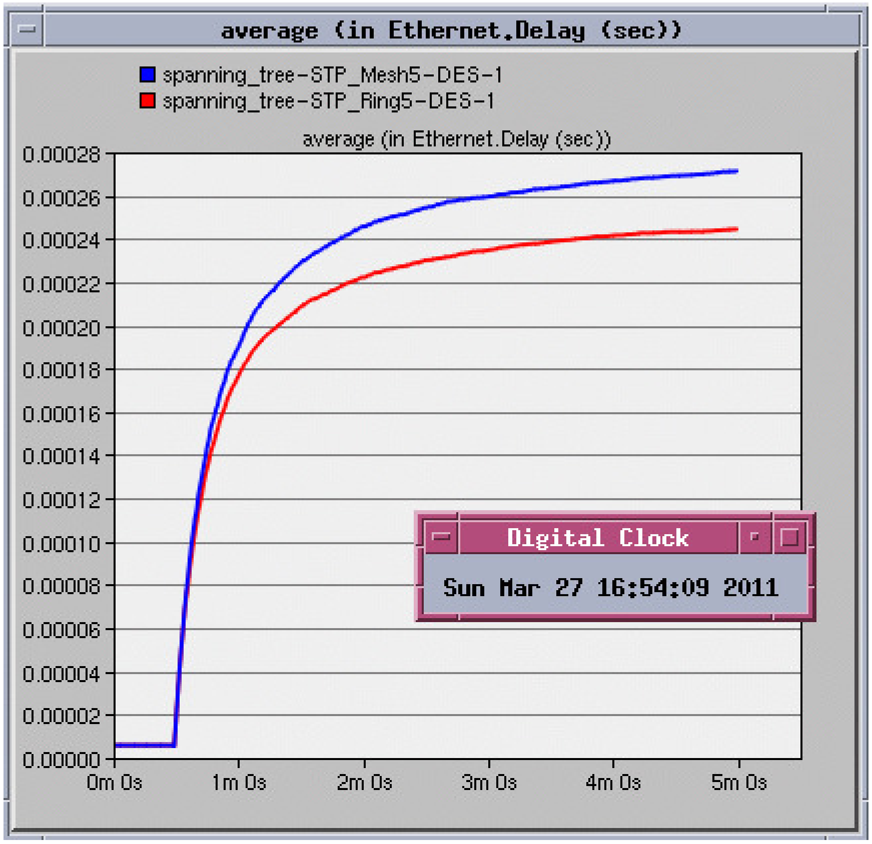

Figure 17: Ethernet delay comparison between Ring and Mesh topology using STP

Further comparisons were made using Failure/Recovery Node Model on Mesh Topology using STP and RSTP. Figure 18 shows point-to-point throughput of Mesh topology using STP. The top graph shows a steady throughput of the Bridge 4 <-> Bridge 5 link up to 2 minutes before the link fails and after 3 minutes when the link recovers. When the link recovers at 3 minutes, it takes 30 seconds for the Spanning Tree Algorithm to converge a tree and forward traffic. The bottom graph shows throughput of the Bridge 5 <-> Bridge 6 link. This link remains blocked and starts forwarding traffic 30 seconds after the other link failed. Therefore, Spanning Tree protocol takes 30 seconds to reform a tree in Mesh Topology.

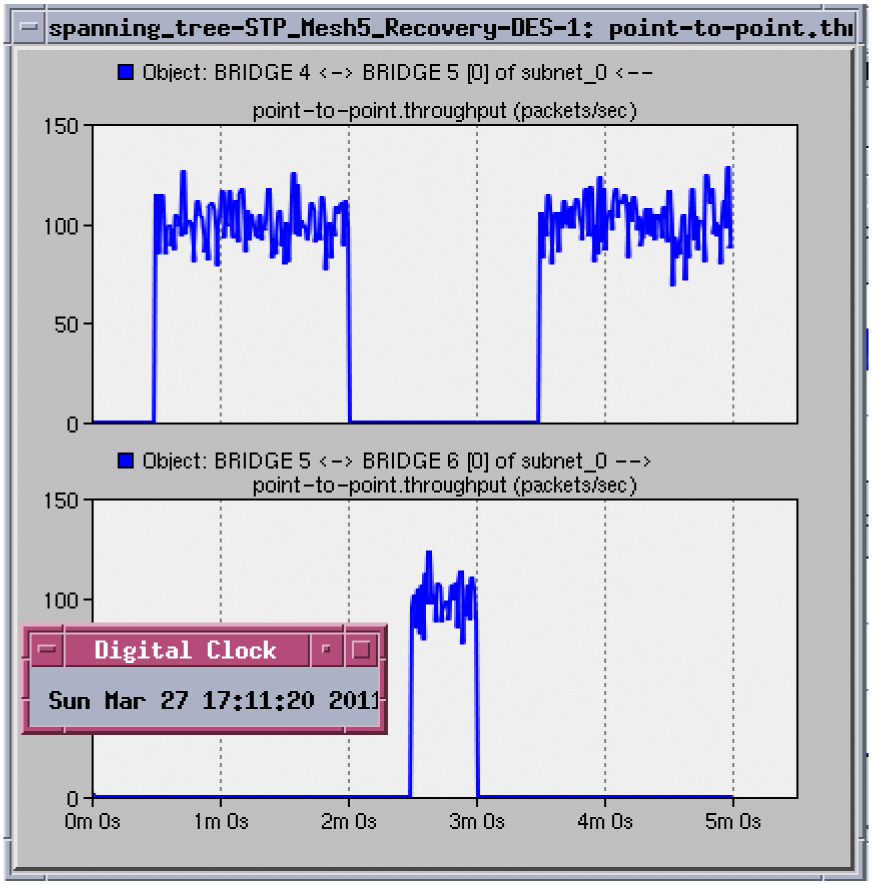

Figure 18: Throughput of Mesh Topology with Failure/Recovery Node Model using STP

A similar analysis was done on Mesh Topology using RSTP as shown in Figure 19. We observed that Rapid Spanning Tree takes 10 seconds to reform a tree in Mesh Topology.

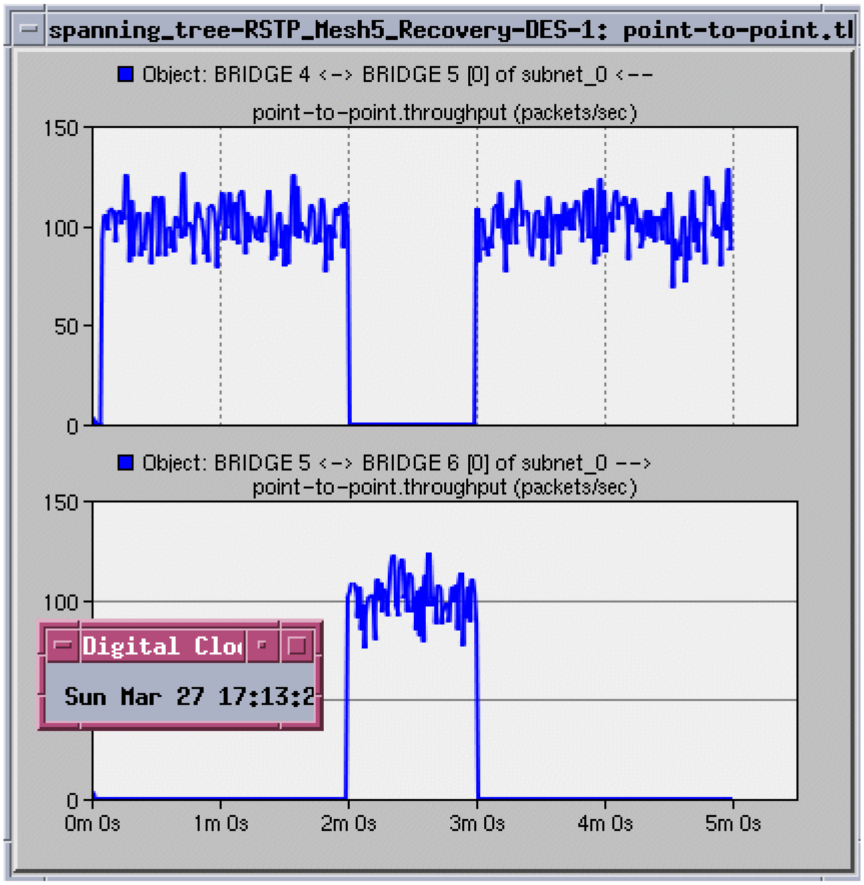

Figure 19: Throughput of Mesh Topology with Failure/Recovery Node Model using RSTP

4.4 Mesh Topology with 8 Bridges

The Mesh Topology used in this scenario was the enhancement of the Mesh topology described in Section 4.3. This enhancement provided us the grounds to compare the spanning tree performance when the number of nodes was increased in a Mesh Topology from 5 bridges to 8 bridges as shown in Figure 20.

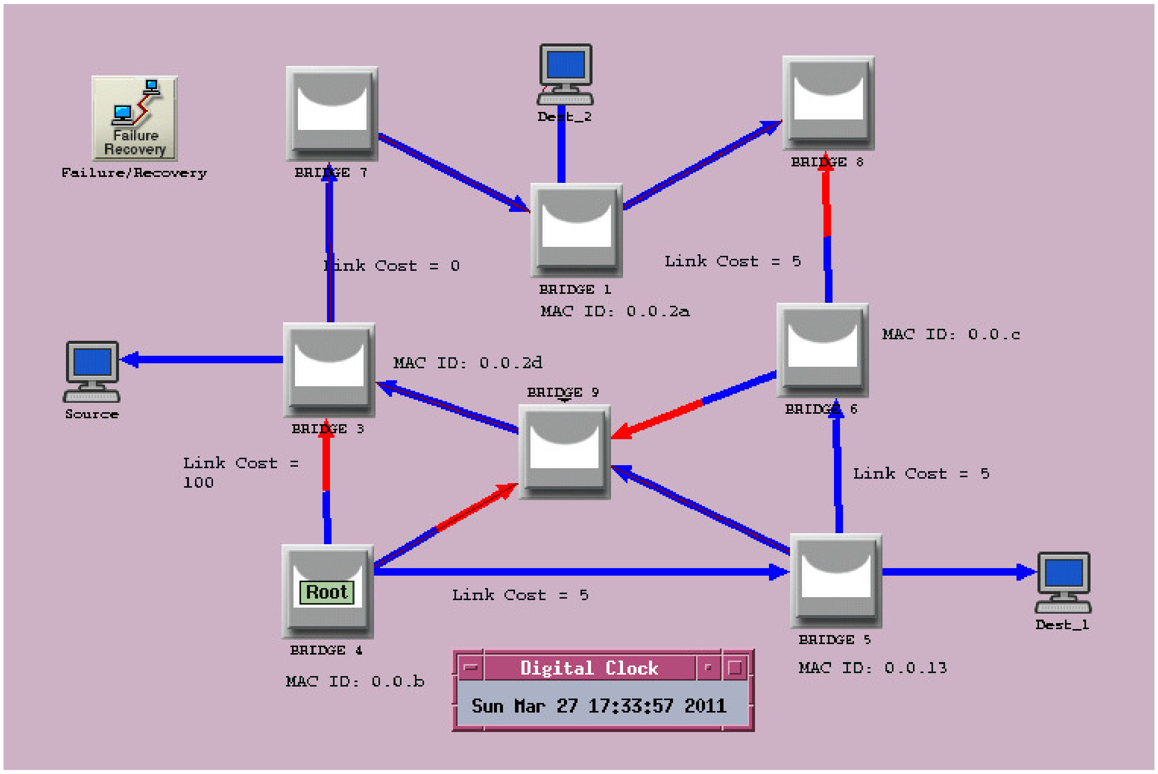

Figure 20: Mesh topology with 8 bridges

4.4.1 Results and Analysis

Figure 21 shows Ethernet delay comparison between Mesh Topology with five bridges versus 8 bridges. As seen from the graph, Ethernet delay with five bridges is 0.28ms whereas Ethernet delay with 8 bridges is 0.25ms. The reason for higher Ethernet delay in 5-bridged topology is - the path cost from source to sink is lesser for 8-bridged topology than the 5-bridged topology.

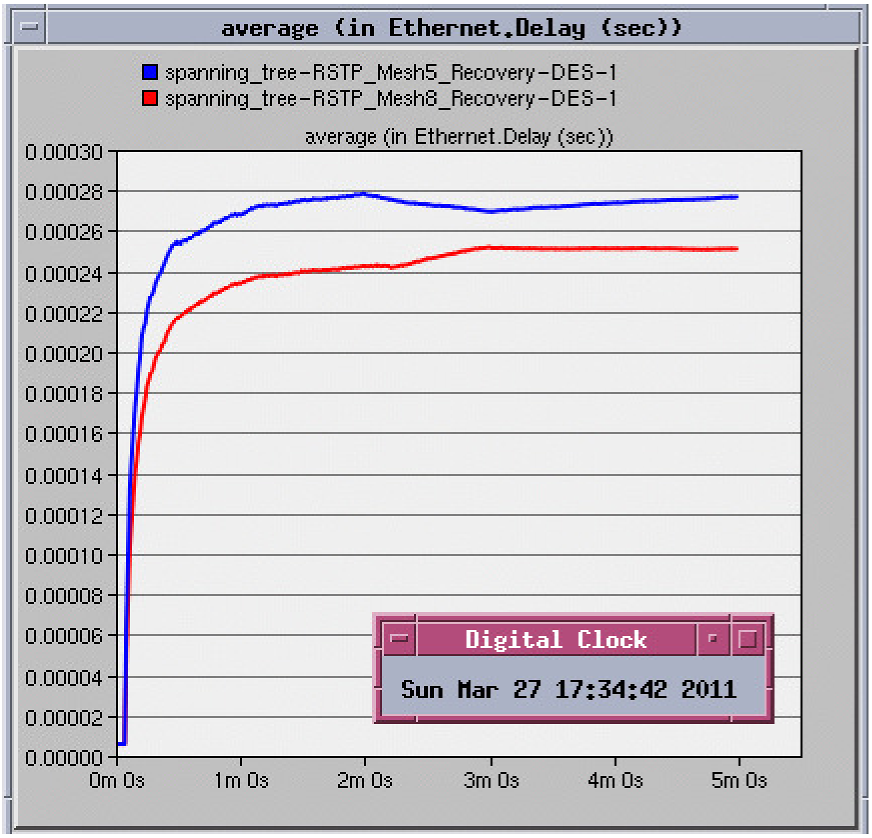

Figure 21: Ethernet delay in 8-bridged Mesh topology versus 5-bridged Mesh topology using RSTP

Similar results were observed in 8-bridged Mesh Topology and 5-bridged Mesh Topology using STP as shown in Figure 22.

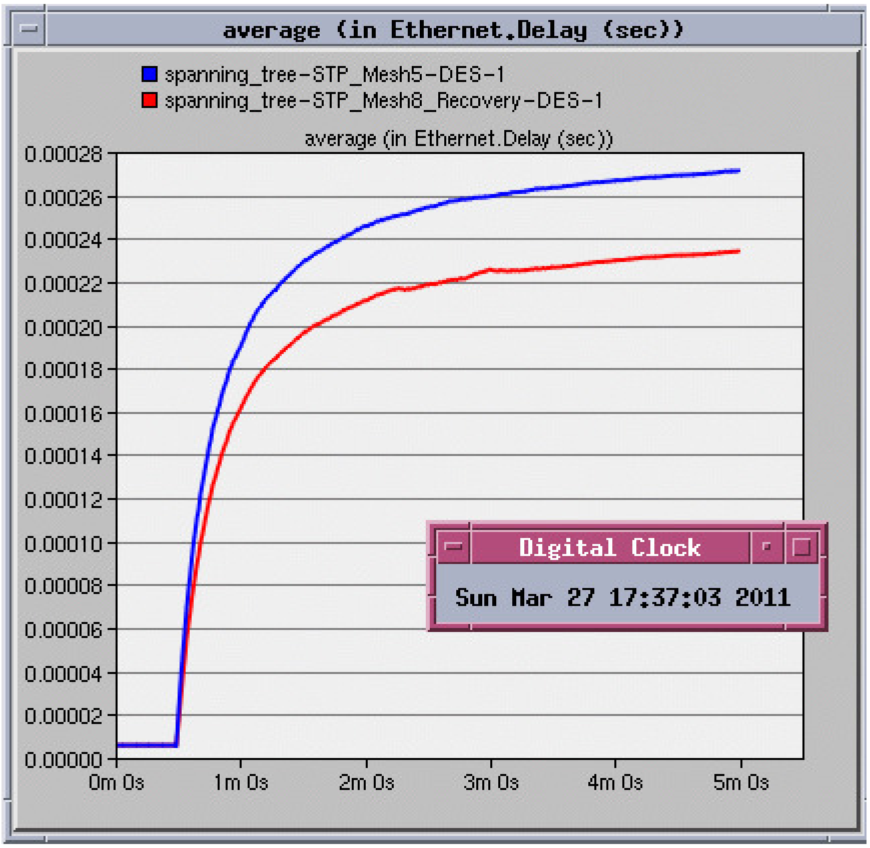

Figure 22: Ethernet delay in 8-bridged Mesh topology versus 5-bridged Mesh topology using STP

Further comparisons were made using Failure/Recovery Node Model on Mesh Topology with eight bridges, using STP and RSTP. Figure 23 shows point-to-point throughput of 8-bridged Mesh topology using STP. The top graph shows a steady throughput of the Bridge 4 <-> Bridge 5 link up to 2 minutes before the link fails and after 3 minutes when the link recovers. When the link recovers at 3 minutes, it takes 30 seconds for the Spanning Tree Algorithm to converge a tree and forward traffic. The bottom graph shows throughput of the Bridge 4 <-> Bridge 9 link. This link remains blocked and starts forwarding traffic 41 seconds after the other link failed. Therefore, Spanning Tree protocol takes 41 seconds to reform a tree in Mesh Topology.

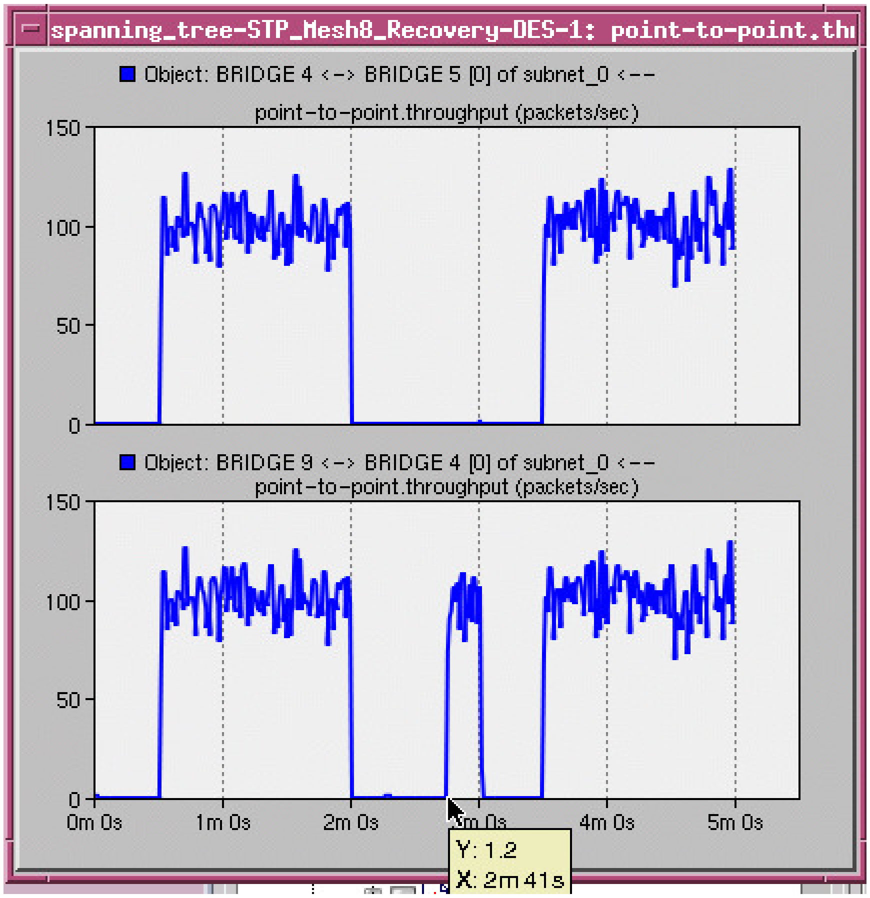

Figure 23: Throughput of 8-bridged Mesh Topology with Failure/Recovery Node Model using STP

A similar analysis was done on 8-bridged Mesh Topology using RSTP as shown in Figure 24. We observed that Rapid Spanning Tree takes 10 seconds to reform a tree in Mesh Topology.

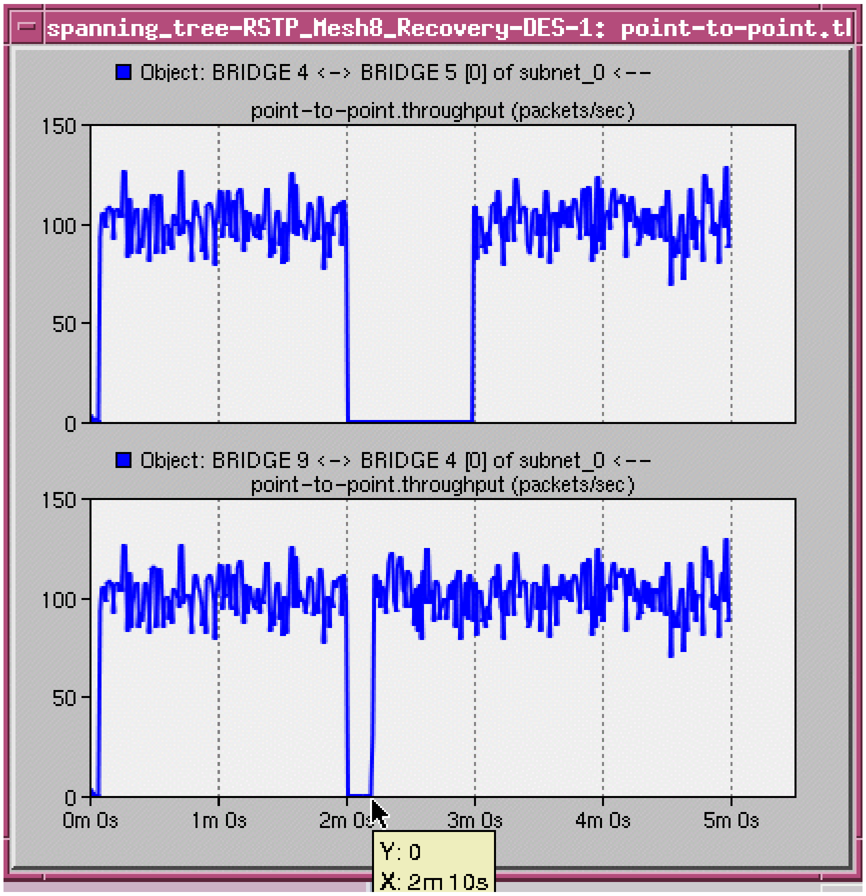

Figure 24: Throughput of 8-bridged Mesh Topology with Failure/Recovery Node Model using RSTP

5.0 Conclusion

Based on the spanning tree protocol chosen, the bridges exchange BPDU frames to determine the root bridge and form a spanning tree. This was observed by running an animation on the OPNET simulator. In the animation, we could see the BPDU frames are sent and received by the bridges and the traffic is sourced by the server and sunk by the workstations. As soon as the network is configured, the bridges determine their root bridge. The following scenarios were modeled and simulated in this project:

- Ring topology with five bridges using STP
- Ring topology with five bridges using RSTP
- Ring topology with five bridges using STP with failure/recovery node model

- Ring topology with five bridges using RSTP with failure/recovery node model
- Mesh topology with five bridges using STP
- Mesh topology with five bridges using RSTP
- Mesh topology with five bridges using STP with failure/recovery node model
- Mesh topology with five bridges using RSTP with failure/recovery node model
- Mesh topology with eight bridges using STP with failure/recovery node model
- Mesh topology with eight bridges using RSTP with failure/recovery node model.

Some of the major differences between STP and RSTP learnt in this project are:

- RSTP has two additional port designations – Alternate Port, acting as a backup for root port and Backup Port, acting as a backup for designated port on a LAN
- In STP, the root bridge triggers the BPDUs whereas in RSTP, all the bridges send BPDUs
- In STP, bridges wait for the time-out of BPDU parameters before changing port states whereas in RSTP, states are changed immediately with change in topology.

From the results and analysis of the scenarios, we concluded:

- STP and RSTP create virtual network and prevents switching loop
- In Ring Topology, STP takes ~30s to form a spanning tree and RSTP takes ~6s to form a spanning tree
- In Mesh Topology, increasing the number of links did not change the spanning tree formation time, however, increasing the number of nodes increased STP formation time by 11s and RSTP formation time by 4s
- RSTP uses more BPDU traffic than STP

6.0 Organization and Timelines

We started the project work from the 3rd week of the semester and the first month was spent understanding the spanning tree protocols and Data Link layer components. Throughout the semester, we completed and learnt OPNET tutorials which helped us in modelling and simulating our project. Table 4 gives a split of the tasks and time it took to complete each milestone.

Table 4: Task division and Timelines of the project

Task	Completed by	Completion Time
Understand Spanning Tree protocols	Simran and Manjur	4 weeks
Familiarize with OPNET14.0	Simran and Manjur	12 weeks
Create Ring Model	Simran and Manjur	4 weeks
Create Mesh Model with 5 bridges	Simran and Manjur	1 week
Create Mesh Model with 8 bridges	Simran and Manjur	1 week
Analyze results	Simran and Manjur	2 weeks

7.0 Future Work

We learnt that there is a maximum age associated with each BPDU frame which specifies the maximum expected arrival time of hello BPDUs. If the maximum age timer expires, the bridge detects that the link to the root bridge has failed and initiates a topology re-convergence. Our future work would include determining the maximum number of nodes supported by a tree to ensure the BPDU travels to all the nodes before its maximum age expires.

Also, we noticed that after a link failure, the Spanning Tree Algorithm calculates the new spanning tree which takes about 6 seconds in RSTP and 30 seconds in STP. A possible solution to reduce this calculation time is by storing the pre-calcuated spanning trees with every possible link failure in the given network. This solution will improve the efficiency of time sensitive, real-time network connections.

8.0 References

[1] H. Forbes, "Planning for Converged Plantwide Ethernet," *ARC Brief Advisory Group,* June 3, 2010

[2] G. Wang, J. Liu, L. Wu and H Yao, "Three-Rings Redundancy Industrial Ethernet Based on RSTP," *International Conference on Signal Processing Systems,* Singapore, pp 228-231, May 2009

[3] D. DesRuisseaux and S. Electric, "Use of RSTP to Cost Effectively Address Ring Recovery Applications in Industrial Ethernet Networks," *ODVA 2009 Conference and 13th Annual Meeting,* Florida, USA, February 2009

[4] Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network," *Proc. of IEEE Infocom 2005*, vol. 3, 13-17, pp. 2223-2234, March 2005

[5] H. S. Chiu, B. Wu, K. L. Yeung, and K.-S. Lui, "Widest Spanning Tree for Multi-Channel Multi-Interface Wireless Mesh Networks," *Proc. Of IEEE WCNC*, pp. 2194-2199, April 2008

[6] K. Lui, W.C. Lee and K. Nahrstedt, "STAR: A transparent spanning tree bridge protocol with alternate routing," *ACM SIGCOMM Computer Communications*, Review 32, July 2002

[7] J. Farkas and Z. Arato, "Performance Analysis of Shortest Path Bridging Control Protocols," *Global Telecommunications Conference,* Budapest, Hungary, pp.1-6, Nov./Dec. 2009

[8] A. Azcorra and G. Ibanez, "Application of Rapid Spanning Tree Protocol for Automatic Hierarchical Address Assignment to Bridges," *Telecommunications Network Strategy and Planning Symposium,* Madrid, Spain, pp. 435-440, June 2004

[9] S. Angelescu, "Chapter -4: Spanning Tree Protocol," *CCNA Certification All-In-One for Dummies ,* 5th Edition, pp. 385-408.

[10] J. T. Yu, "Applying IEEE 802.1w (RSTP) to Improve Service Availability," *IEEE International Conference on Dependable Systems and Networks,* pp. B10-B11, June 2003.

[11] J. T. Yu, "Performance Evaluation on Linux Bridge," *Telecommunications System Management Conference*, Louisville, Kentucky, April 2004

[12] P. Radia, "An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN," *ACM SIGCOMM Computer Communication Review,* pp. 44–53, 1985.

[13] P. Radia, *Interconnections, Second Edition*. USA: Addison-Wesley. ISBN 0-201-63448-1, 2000

[14] http://www.s3.kth.se/lcn/courses/2E1623/pdf/STP.pdf

[15] Cisco,“ Understanding Rapid Spanning Tree Protocol (802.1w),” *Document ID: 24062*