McGill University - Faculty of Engineering
Department of Electrical and Computer Engineering

ECSE–494 – Telecommunication Networks Lab
Prof. M. Coates
Winter 2003

Experiment 5: LAN Operation, Multiple Access Protocols and IP routing Protocols

Antoine Lourtau 119921493
Gregory Zoughbi 110021100

Table of Contents

Abstract……………………………………………………………………….3

1. Introduction……………………………………………………………….3
1.1. Ethernet MAC 3
1.2. Token Ring 3
1.3 OSPF (Open Shortest Path First) 4

2. Methodology…………………………………………………………………4
2.1. Ethernet Performance for increasing hop propagation delays 5
2.2. Token Ring Performance for varying hop propagation delays 5
2.3. Ethernet Performance for varying traffic loads 5
2.4. Token Ring Performance for varying traffic loads 5
2.5. OSPF 6

3. Results and Discussion……………………………………………………6
3.1. Ethernet Performance for increasing hop propagation delays 6
3.2. Token Ring Performance for varying hop propagation delays 7
3.3. Ethernet Performance for varying traffic loads 7
3.4. Token Ring Performance for varying traffic loads 8
3.5. OSPF 9

Conclusion…………………………………………………………………....9

References………………………………………………………………………10

Abstract

The following experiment serves a twofold purpose: to investigate the mechanisms proprietary to LAN operations and multiple access protocols, and to explore the workings of IP routing protocols. In particular, we begin by assaying the performance of Ethernet and Token Ring network ring architectures by varying hop propagation delays and traffic load. We then examine the performance and behavior of IP routing using OSPF (Open Shortest Path First) routing.

1. Introduction

1.1 Ethernet MAC

Ethernet is the most commonly used technology to build Local Area Networks (LANs). Ethernet MAC (Media Access Control) is the portion of Ethernet core that handles the CSMA/CD (carrier sensing multiple access with collision detection) protocol for transmission and reception of frames. It performs Frame Data Encapsulation and Decapsulation, Frame Transmission, and Frame Reception. MAC provides the link layer for the Ethernet LAN system [1].

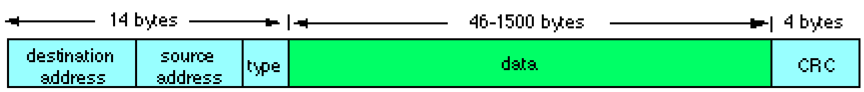

Figure 1.1: MAC encapsulation of a data packet

The MAC encapsulation consists of three parts (*figure 1.1*): the header (itself comprised of the destination address, the source address and the type), the data, and CRC (cyclic redundancy check) for error checking. CSMA improves performance of Ethernet in high medium utilization scenarios. When a node has data to transmit, the node first listens to the cable (using a transceiver) to see if a carrier (signal) is being transmitted by another node [2]. However, this alone is unable to prevent two nodes from transmitting at the same time. If two nodes simultaneously try transmitting, then both could see an idle physical medium, and both will conclude that no other node is currently using the network, and both will then decide to transmit and a collision will occur. The collision will result in the corruption of the data being sent, which will subsequently be discarded by the receiver since a corrupted Ethernet frame will almost surely not have a valid 32-bit MAC CRC at the end.

1.2 Token Ring

Token Ring was initially developed by IBM as a protocol to enable reliable implementations of LANs using a ring architecture. In order to avoid collisions over the network, the Token Ring protocol uses a "token" (a control permission) to grant or relinquish access to the network. The data packets are sent from one node to another, and only the node in possession of the token is allowed to transmit, thus avoiding collisions. Each station repeats the data, checks for errors, and copies the data if appropriate, and when the data is returned to the sending node, the packet is discarded [3]. The reliability of Token Ring lies in the fact that "dead" or broken nodes can be bypassed, thus making the overall system very fault tolerant. Unlike CSMA/CD networks (such as above described Ethernet), Token Ring networks are *deterministic*, which means that it is possible to calculate the maximum time that will pass before any end station will be capable of transmitting. This feature and several efficient reliability features make Token Ring networks ideal for applications in which delay must be predictable and robust and reliable network operation is important [4].

1.3 OSPF (Open Shortest Path First)

OSPF is a link-state routing protocol. It is designed to be run internal to a single Autonomous System (AS). Each OSPF router maintains an identical database describing the Autonomous System's topology. From this database, a routing table is calculated by constructing a shortest-path tree. OSPF recalculates routes quickly in the face of topological changes, utilizing a minimum of routing protocol traffic. OSPF provides support for equal-cost multipath. An area routing capability is provided, enabling an additional level of routing protection and a reduction in routing protocol traffic. In addition, all OSPF routing protocol exchanges are authenticated [5]. OSPF can operate within a hierarchy, where the largest entity within the hierarchy is the AS (a collection of networks under a common administration that share a common routing strategy). OSPF is an intra-AS (interior gateway) routing protocol, although it is capable of receiving routes from and sending routes to other ASs. OSPF is based on the SPF (Shortest Path First) algorithm, also known as Dijkstra's Algorithm. Thus, each router periodically sends an link-state advertisement (LSA) to provide information on a router's adjacencies or to inform others when a router's state changes. By comparing established adjacencies to link states, failed routers can be detected quickly, and the network's topology can be altered appropriately. From the topological database generated from LSAs, each router calculates a shortest-path tree, with itself as root. The shortest-path tree, in turn, yields a routing table [6].

2. Methodology

This experiment was carried out using the OpNET simulation software. Both parts required us preloading scenarios for the LAN operations and IP routing experiments. For the LAN experiment, both the Ethernet and Token Scenarios are based on a star-like topology, with a central router. The OSPF experiment is based on 4 Autonomous Systems (ASs) connected together via routers, as depicted in *figure 2.1*. The link speeds vary between copper DS3 lines, and OC3 links.

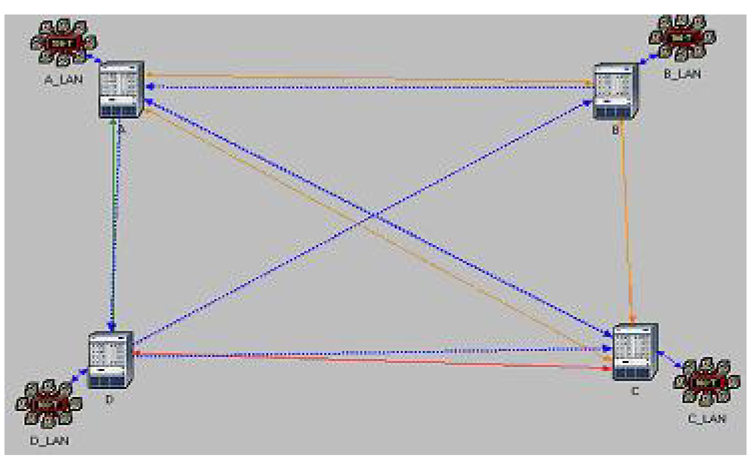

Figure 2.1: OSPF experiment network topology

2.1 Ethernet Performance for increasing hop propagation delays

We select the Ethernet scenario from LAN_Exp5. The default settings set the network simulation to be carried out with a traffic that is exponentially distributed with a mean of 0.0004, a packet size exponential set to 1024, a ON state exponential set to 10, and an OFF state exponential set to 90. We simply verified that these values were properly assigned to each node, by viewing the attributes for each link. We then proceeded to increase the hop propagation delay to 50us, 1ms, 2 ms, 5ms, and 500ms by creating individual scenarios for each delay value, in order to display the comparative graphs. While running the experiment, it was somewhat visible that fluctuating the hop propagation delay between 0ms and 5 ms would be sufficient to view the behavioral trend, however, we added the half second delay just to confirm our prognostics. We performed the simulations over a period of 3 minutes, with a default seed value of 128, and graphed the utilization of the links (as a global parameter, time averaged). However, for certain simulations, we modified this seed value in the hopes of compensating for the undesirable behavior of the random traffic generation (some results were slightly unexpected because of this).

2.2 Token Ring Performance for varying hop propagation delays

We select the Token Ring scenario from LAN_Exp5. By default, the settings for the network's configuration are identical to those of the Ethernet scenario described in section 2.1. The only difference is the mean of the exponential distribution that is set to a value of 0.002. In addition, by default, the propagation delay on the links is 3.3E-6s, whereas it was "distance-based" in the Ethernet experiment. We then proceed to vary the hop propagation delays. On one hand we created a scenario for a decreased delay of 1us, and on the other we create scenarios for 1ms, 2ms, 5ms and 100ms delays. Again, it was quite visible that increasing the delay beyond 100ms would prove to be futile within the scope of this experiment. We then executed the simulation over a period of 3 minutes, and displayed the utilization as a timed average.

2.3 Ethernet Performance for varying traffic loads

Here we select the original Ethernet scenario. We keep the configurations as they are set, besides setting the link delay to 3.3E-6s. We then proceed to varying the traffic load by modifying the mean of the exponential distribution function. Indeed, by decreasing the mean value we can increase the amount of traffic generated. Therefore, we create 3 more scenarios with means values of 0.0003, 0.00025, and 0.0002. We did not increase that traffic beyond that value since the simulations were already taking substantial amounts of time to process, and the sought trend had been identified. We then plotted both the utilization of the links and the generated traffic over a period of 2 minutes.

2.4 Token Ring Performance for varying traffic loads

Again, we chose the original Token Ring scenario, from which we create 4 similar scenarios with different mean values for the exponential distribution function. Starting with the default value of 0.002, we decrease the mean value in each scenario to 0.0013, 0.0012, 0.0010, and 0.0005. We then plot both link utilizations and traffic throughput over 2 minutes.

2.5 OSPF

For this part of the experiment we use the project saved as OSPF_exp5. This project contains two scenarios. The first step is to determine the time required before the routing algorithm converges. To do so we run the simulation for 10 minutes, and obtain the OSPF Traffic Sent parameter to be graphed over a period of time. This allows us to read the time required before the routing algorithm converges. We then seek to view the routes assigned by OSPF to the various traffic demands. To do so, we select the ***IP Demands*** option under the ***Protocols*** menu, and specify ***Display Routes***. We then pick the flows that we wish to visualize and enable the display option.

We then proceed to determine the most heavily used link, and plot the utilizations of each link, overlapping each plot.

In order to minimize the peak utilization, we upgrade the capacity of the most heavily used link.

We then proceed to assigning explicit link costs, we edit the attributes of the Router, select the ***OSPF Parameters***, view the ***Interface Information***, select a ***row***, and set the ***cost*** explicitly. The row number corresponds to the interface or port number of each element connected to it. This information is viewed by selecting the attributes for each element of the network, and reading the interface number.

We then fail the link B→C, and repeat the procedure to view the routes chosen by the routing algorithm.

Finally, we switch to the other scenario. First we fail the link B→C, observe how the algorithm responds to this failure, and assign the link costs accordingly, as we did above.

3. Results and Discussion

3.1 Ethernet Performance for increasing hop propagation delays

As figure 3.1 illustrates below, better utilization is achieved with smaller propagation delays. The “Ethernet” curve has the highest utilization because it has the lowest delay. As the delay increases, the utilization curve starts becoming less smooth and drops below the ideal zero-propagation-delay curve. To understand why this is so, compare two links that have the same throughput, but the data stays in the links different amount of times. The less time the data stays in the link, the more efficiently the link is utilized. As the delay becomes in the order of milliseconds, the link utilization becomes very low. We also notice that due to the random traffic generator used in OPNET, several curves of comparable propagation delays might have fluctuations that can cause the curve with the higher delay to appear to have higher utilization in some period of time (see 1ms and 2ms delays over 1m25s – 2m).

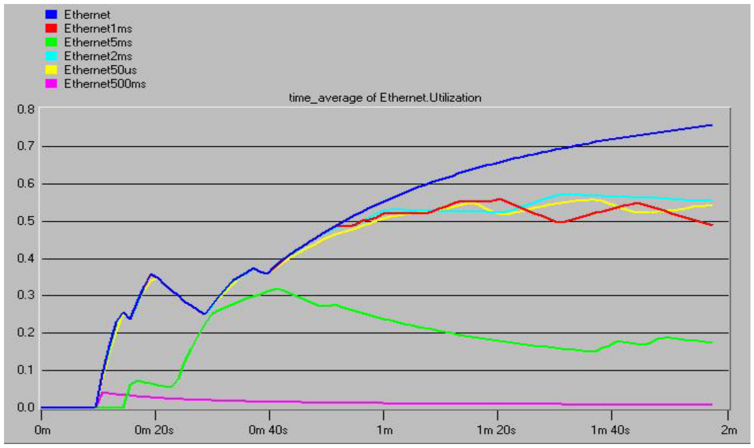

Figure 3.1: Time average of Ethernet utilization as a function of propagation delay.

3.2 Token Ring Performance for varying hop propagation delays

Despite behaving similarly when the value of the propagation delay changes (larger delay implies lower utilization), the token ring has significantly lower link utilization. This is due to the structure of the network; each packet is passed along many nodes in the network until it reaches its destination. In our scenario, the maximum utilization value at any time centers around 20% for very low delays. This makes sense since the token ring network has 5 nodes, hence each link can have a maximum utilization of about one fifth. Further, notice that propagation delays that were acceptable in the Ethernet are not acceptable here; 5ms delay achieved 31% utilization for the Ethernet at some point, whereas the maximum it achieved in the token ring was less than 8%.

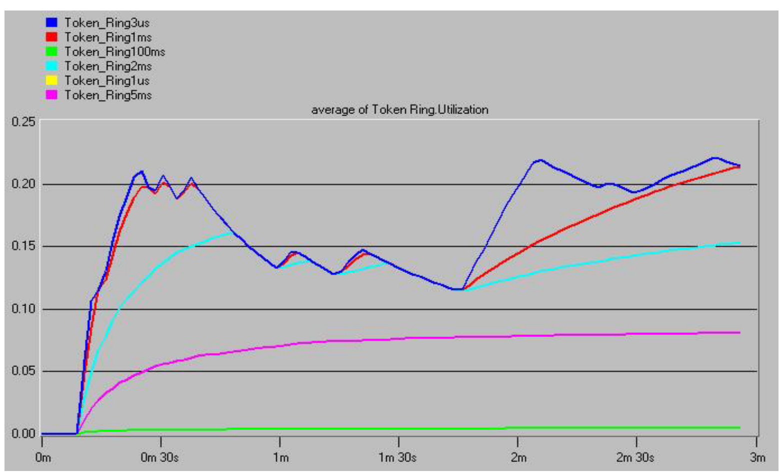

Figure 3.2: Average token ring Utilization as a function of propagation delay.

3.3 Ethernet Performance for varying traffic loads

Figure 3.4 indicates that the traffic sent increases with a larger mean for the exponentially-distributed traffic. Figure 3.3 indicates that the Ethernet delay increases as the average traffic sent increases. Notice that the delay continues to increase even though the traffic remains about the same during the last minute of the simulation. This is because the buffers at the beginning of the links continue to get more full as a result of sending more packets than the links can handle.

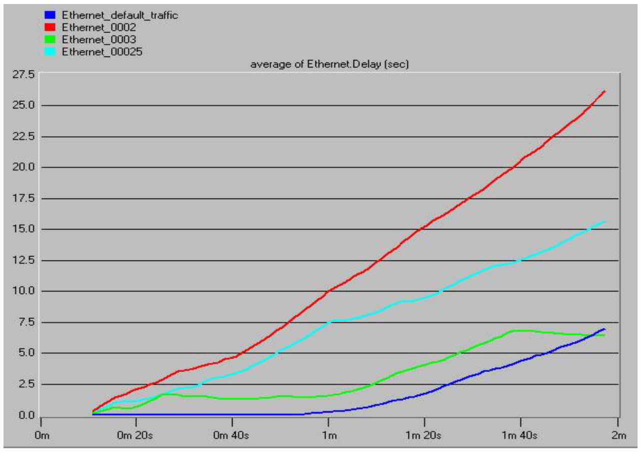

Figure 3.3:Average Ethernet delay as a function of traffic load.

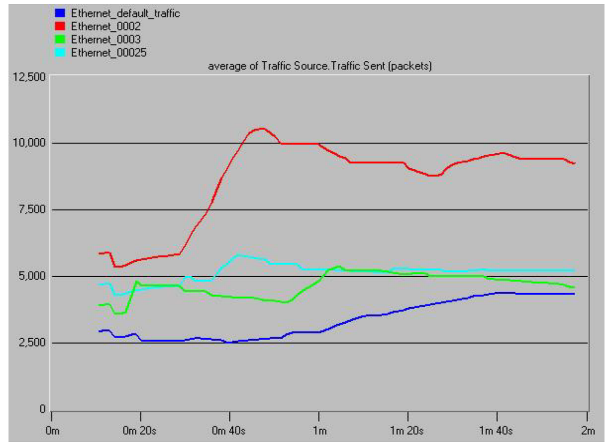

Figure 3.4: Average Ethernet packets sent as a function of traffic load.

3.4 Token Ring Performance for varying traffic loads

The relation between the average delay and the traffic sent in the token ring seems to exhibit similar behaviour as in the Ethernet. However, the delay seems to be more sensitive to variations in the traffic. Furthermore, the delay curve seems to have a shape similar to that of the traffic, and the queuing effects that caused the delays in the Ethernet to continuously increase are not present in the token ring case; the token ring delay seems to converge to a certain value. This is due to the ring structure here rather than the star structure of the Ethernet.

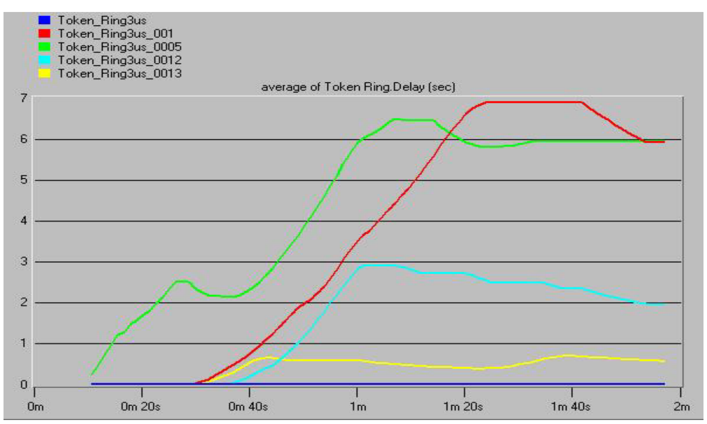

Figure 3.5: Average token ring delay as a function of traffic load.

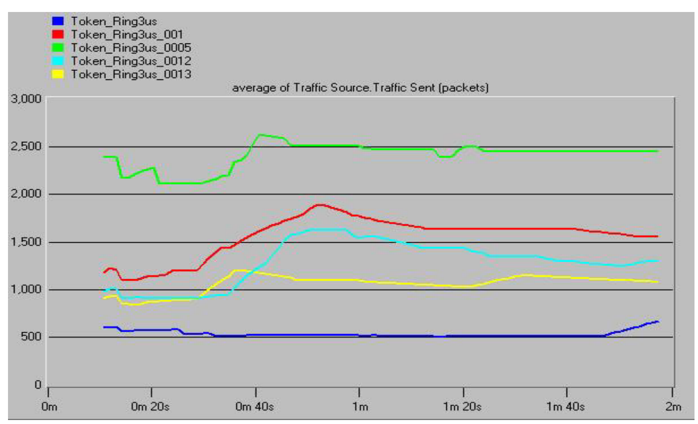

Figure 3.6: Average token ring packets sent as a function of traffic load.

3.5 OSPF

By inspecting the utilization of the links in figure 3.8, w notice that the curves become near constants after about 10 minutes. The routing algorithm is said to converge when the changes in the curves become small.

Figure 3.7 below shows the bursts of the traffic. It consists of small bursts and two large bursts centered after about 20 seconds and 42 seconds. By considering figure 3.8, we notice that the most heavily utilized link after convergence is link A<-->D which utilizes about 82%, whereas all the other links utilize less than 70% each. To minimize peak utilization through the network, we upgrade link A<-->D from DS3 to OC1.

OSPF determines the interface (link) costs by considering the link bandwidths. Links with high bandwidths are assigned low costs, whereas links with low bandwidths are assigned high costs. It should be noted that one cannot achieve better effects by assigning link costs explicitly, as we tried reassigning link costs and the end result was the same. Nevertheless, assigning interface costs can be helpful in other scenarios. For example, suppose link B<-->C has a high chance of failure. In that case, we would assign it a high interface cost so that OSPF would route the traffic around it most of the time. This ensures that the network is not overloaded in case the aforementioned link fails. To simulate this scenario, we assigned link B<-->C a cost of 50, and the rest of the links a cost 5 each. If link B<-->C fails, then the only change in the routing demands is that of the traffic from D to B. Originally, it was routed via C because link A<-->D was the most utilized link. However, after B<-->C fails, the traffic can only be routed via A.

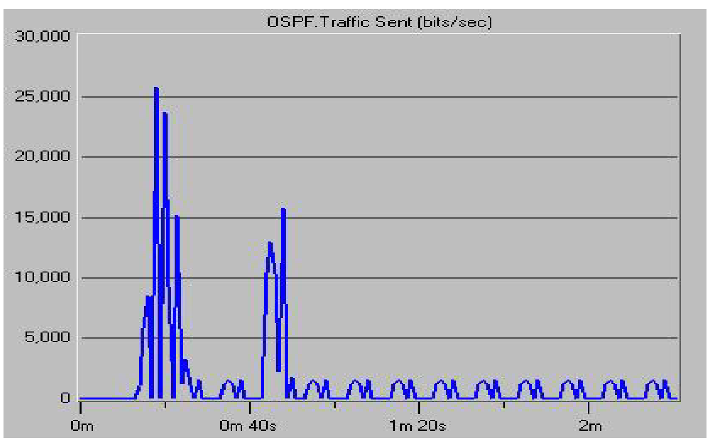

Figure 3.7: Average traffic sent in OSPF.

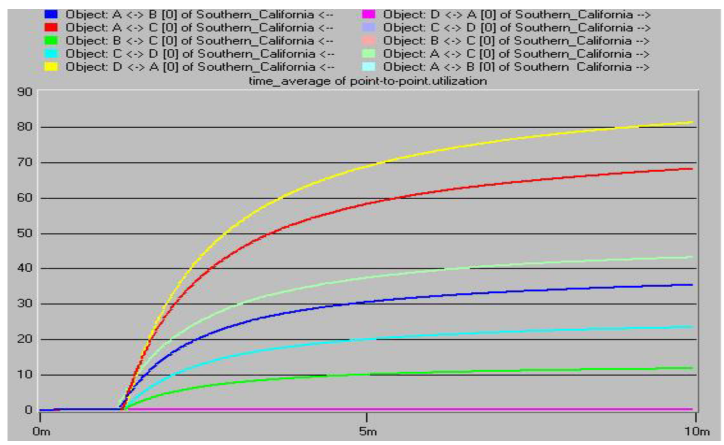

Figure 3.8: Average point-to-point utilization for several links.

4. Conclusion

Ethernet and Token Ring are different approaches to handling the problem of collisions and its effect on performance in LAN implementations. While Token Ring seems to be more sensitive to bulky traffic and considerable propagation delays than Ethernet, its overall performance in normal running conditions is similar to that of Ethernet, with the added advantage of increased robustness. OSPF is an efficient routing algorithm, in particular due to its ability to update routes dynamically in reaction to changes in elements of its hierarchical level.

References

[1] Fairhurst, G. "*Medium Access Control (MAC)*", 2002.
http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/mac.html

[2] Fairhurst, G. "*CSMA/CD*", 2002.
http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/csma-cd.html

[3] Halsall, F., "*Token Ring*".
http://www2.rad.com/networks/1996/toknring/toknring.htm

[4] Cisco Systems, "*Token Ring*", 2002.
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/tokenrng.htm

[5] Moy, J. "*OSPF Version 2*", RFC 2328, 1998.
http://www.ietf.org/rfc/rfc2328.txt

[6] Cisco Systems, "*OSPF*", 2002.
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ospf.htm