The objective of the lab was to simulate an Ethernet Local Access Network (LAN) using OpNet ITGuru. The simulation was of a 10Base5 Ethernet LAN (connected using coaxial cable). The next two sections present the results of each simulation and include an analysis of the network operating under different load conditions. The final section presents conclusions.

Configuration

The simulation was of 30 workstations connected to a 10Base5 Ethernet LAN. Fig. 1 shows the topology of the simulation. The network was configured using the *ethcoax_station* node model, the *eth_coax* link model, and the *eth_tap* tap model.

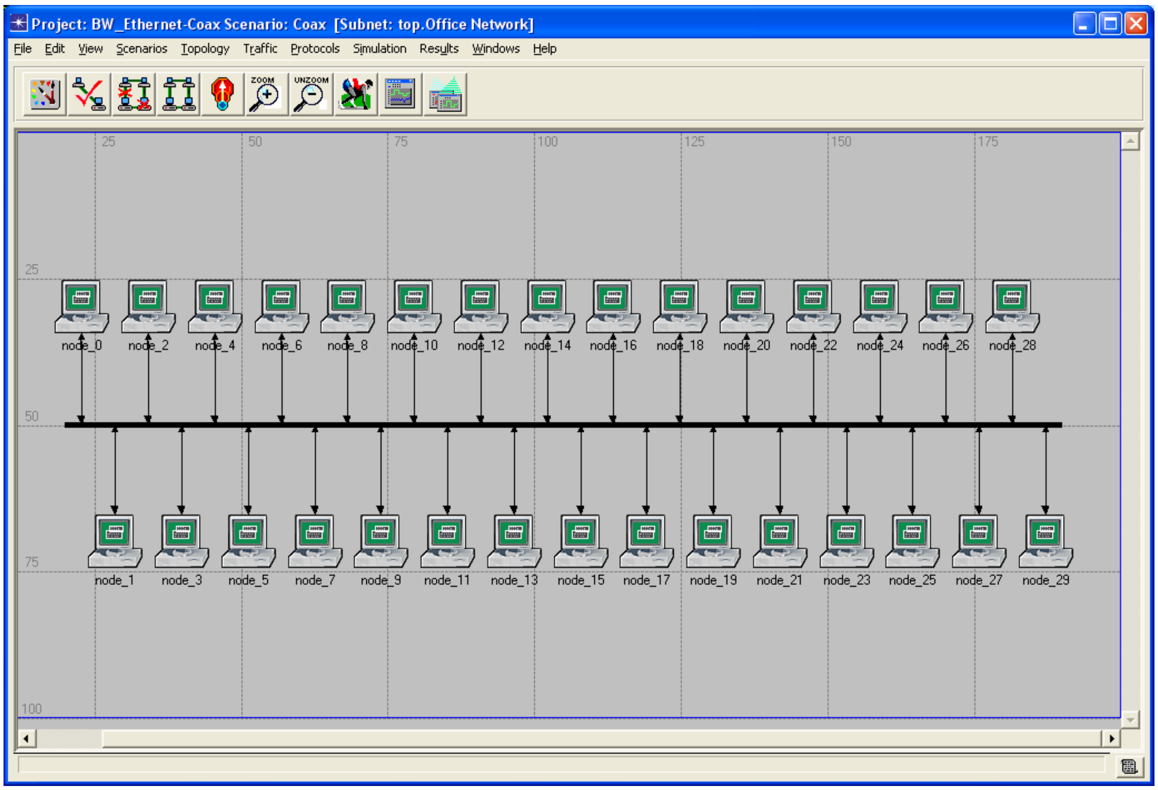

Figure 1 – Ethernet LAN Topology

The LAN link was set to use the *eth_coax_adv* model. Rather than generating extremely large traffic loads, the propagation delay of the Ethernet was increased to 50 ms/m. This allowed the simulation to load the network without increasing the running time of the simulation runs excessively.

All 30 workstations were configured identically. Each was assigned an *On State Time* (time in which it was allowed to send packets) of **exponential(100)**, and an *Off State Time* (dead time between packet transmissions) of **exponential(0)**. The *packet size* was set to **constant(1024)**. The *interarrival time* (time distribution for generation of packets to send) was configured to be one of the following list of nine values (causing a set of nine runs of the simulation):

- **exponential(2.0)**
- **exponential(1.0)**
- **exponential(0.5)**
- **exponential(0.25)**
- **exponential(0.1)**
- **exponential(0.05)**
- **exponential(0.035)**
- **exponential(0.03)**
- **exponential(0.02)**

The simulation was set to capture the network-wide *packets received / sec* and *packets sent / sec* for each of the nine simulation runs. For each run, it captured a single scalar value that represented the time-average of the statistic over the duration of the run.

Running the Simulation

The simulation was executed for 15 seconds for each run. Each run represented a progressively greater volume of traffic offered to the network. This resulted in progressively longer execution times for each run, ranging from less than one second for the first three runs to 55 seconds for the last run.

After the simulation was completed, a graph was generated showing the rate of packets received (on the vertical axis) vs. the rate of packets sent (on the horizontal axis). This represents the network thoughput vs. network load. These values were the time-averages over the duration of each run. Fig. 2 shows the graph that was generated.

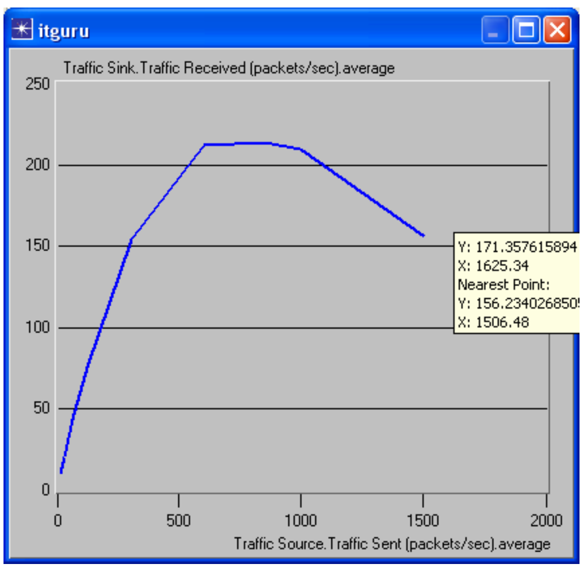

Figure 2 – Packets Received vs. Packets Sent

The graph shows that as the volume of traffic offered to the network increased, the volume of traffic successfully delivered increased roughly linearly, up to a point. It then reached a plateau, and as offered traffic increased, the delivery rate actually decreased. This reflects the fact that increasing demand for a CSMA/CD network causes increasing collision rates, which in turn decreases link throughput.

Additional Scenarios

Three additional simulation scenarios were created; each of the three had the same 30 nodes, but the pattern for traffic offered to the network was changed to be a single value (so only one run of each simulation was made). The scenarios, and their *interarrival time* distributions, are as follows:

- Coax_Q2a - **exponential(0.1)**
- Coax_Q2b - **exponential(0.05)**
- Coax_Q2c - **exponential(0.025)**

The network-wide traffic received, in packets/sec, was collected for each of these scenarios. In addition, the collision count was collected for one workstation on the network. Fig. 3 shows the time-average value of the collision count for each scenario, and Fig. 4 shows the time-average and instantaneous value of the network throughput (packets/sec received).

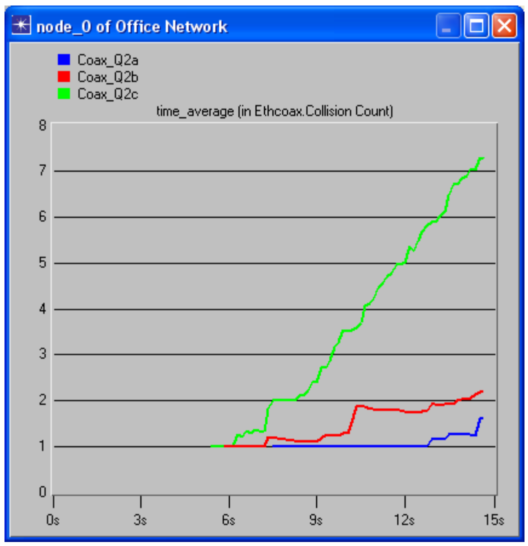

Figure 3 – Time Average Collision Count for Each Scenario

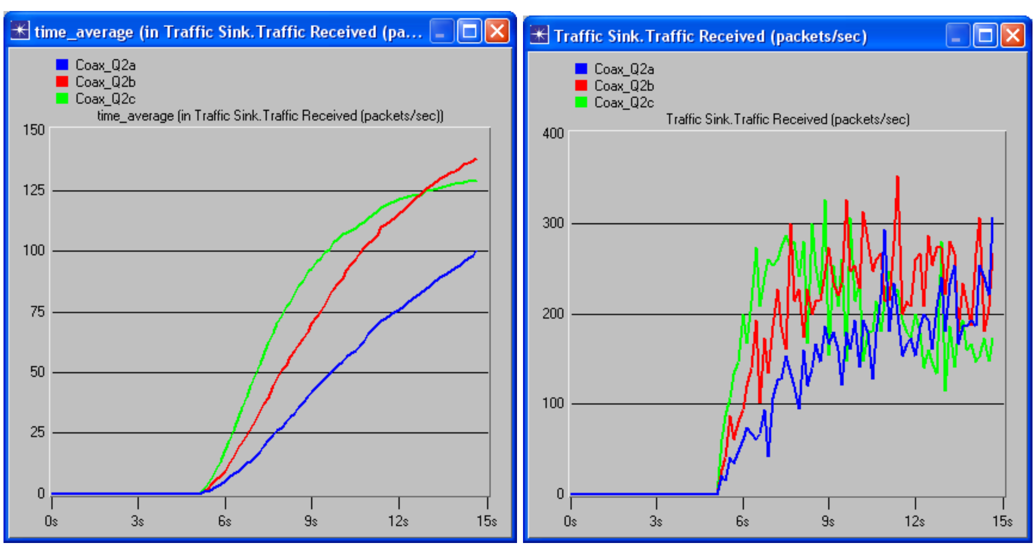

Figure 4 – Network Throughput for Each Scenario

This shows that as the load offered to the network increases, the collision count increases. The difference between scenarios **Coax_Q2a** and **Coax_Q2b** is not too significant; there are more collisions with the higher offered load, but the network appears stable. However, for scenario **Coax_Q2c**, the collision count is rising rapidly as time goes forward. The load offered to the network is too large for it to handle, and the operation of the Ethernet causes it to degrade over time. When a packet transmission finishes, if multiple nodes need to send, they will all try to send immediately (because of the *1-persistent* Ethernet transmission algorithm). With heavier loads, the chance of a collision immediately following the end of a packet increases, so more of the available bandwidth is consumed by collisions. This lowers overall network throughput. This can be seen in the instantaneous throughput in Fig. 4.

An additional scenario, **Coax_Q3**, was created by copying **Coax_Q2c** and removing half the workstations from the LAN. It was also run for 15 seconds. Fig. 5 shows the time-average collision count for one node on the network, comparing the **Coax_Q3** and **Coax_Q2c** scenarios.

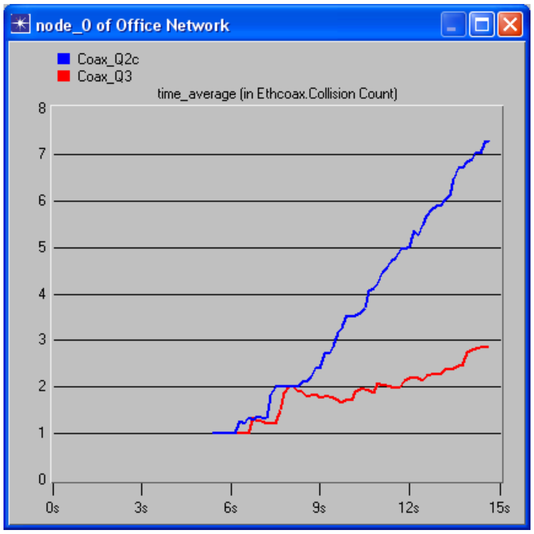

Figure 5 – Comparison of Collision Rate with Smaller Network

As expected, even with the same load being offered to the network by each workstation, with a smaller number of workstations, the collision rate is lower. The load offered by 15 workstations seems to be within the capacity of the network; the **Coax_Q3** scenario appears to be fairly stable.

Finally, another scenario, **Coax_Q4**, was created by copying **Coax_Q2c** and changing the packet size from **constant(1024)** to **constant(512)**. This scenario and **Coax_Q2c** were changed to collect the traffic received in both bits/sec and packets/sec. Both scenarios were run for 15 seconds. Fig. 6 shows the network throughput for the two scenarios. These results show the throughput in bits/sec for scenario **Coax_Q4** were somewhat greater; although less traffic was being offered to the network, fewer collisions resulted in higher net throughput. In terms of packets/sec, **Coax_Q4** had significantly better throughput – over double that of scenario **Coax_Q2c**.

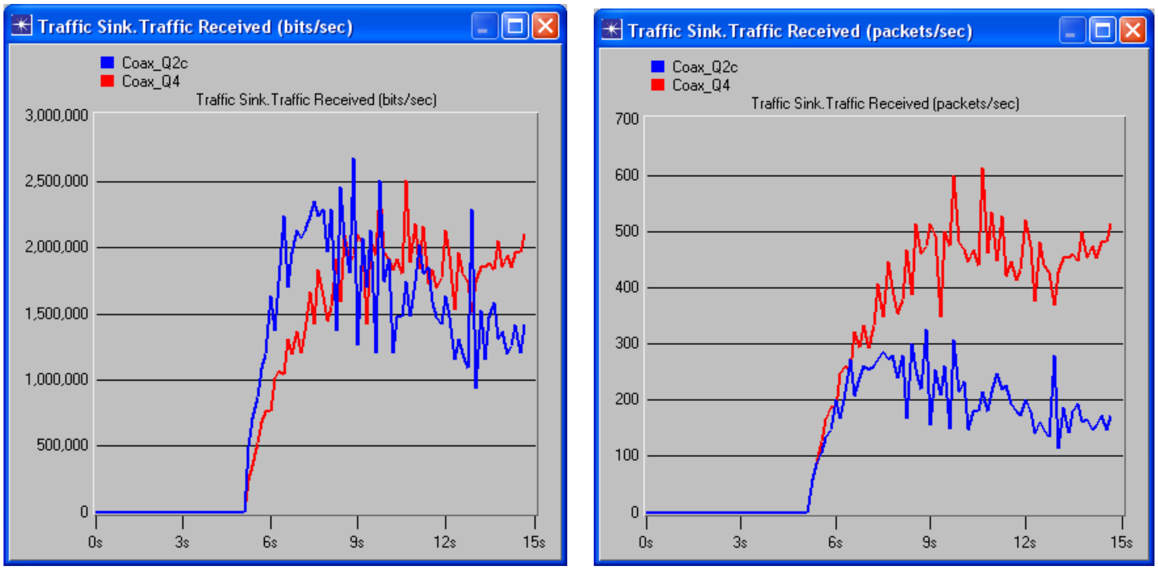

Figure 6 – Comparison of Throughput for Smaller Packet Size – Bits/Sec and Packets/Sec

Conclusions

The different simulation scenarios really illustrated the problems that can be encountered when an Ethernet LAN is overloaded. Not only does network throughput reach a peak, but it actually starts to degrade as load increases. It would be important to consider this when designing an Ethernet LAN; the number of nodes on the LAN would need to be limited based on the anticipated network load they would generate. The scenarios demonstrated that the anticipated load must take into account the frequency of traffic and also the expected average packet size. If more nodes had to be connected to the network, an option would be to create multiple LANs and connect them with a bridge.

The OpNet simulator works fairly well, but some bugs were encountered during execution of the lab. The most significant is that even though all the workstations on the LAN were selected and the Apply Changes to All Selections box was checked, it did not always set the parameter correctly for all workstations. It was necessary to do a lot of double-checking to make sure things were set properly. It is also necessary to have some sort of expectation for what the results should be, so that you can recognize potential problems and troubleshoot them.