Evaluation of some Fundamental Characteristics in an Ethernet LAN through ns2 Simulation

Ayon Chakraborty and Sphoorti Kundargi
Department of Computer Science, State University of New York at Stony Brook.
{aychakrabort, skundargi}@cs.sunysb.edu

I. INTRODUCTION

In this project we explored various characteristics of an Ethernet LAN and studied its dependency on different network parameters like network size, offered load, the size of the transmitted packets, node’s buffer size etc. The fundamental characteristics which we wanted to capture in our experimentations include Throughput, Delay, and Drop Probability among others.

Let’s explain the different characteristics we studied. We studied the aggregated throughput, which is the raw number of data bytes received at the receiver’s end per unit time. The Delay characteristics include two types of delays: The first one was the Network Delay, which means the amount of time that elapsed between dequeuing a packet and it being received successfully. The second was the total delay which also took into account the queuing delays or waiting delays in the node’s buffer. Drop probability is simply the probability of a packet drop. This can be easily computed from the total number of packets put into the wire and the fraction of it that got dropped. We also studied the effect of the node’s buffer size on this drop probability.

In case of very high network loads the Ethernet protocol tends to prefer nodes with lower value of backoff counters, because they try to access the media more frequently in a shorter gap of time compared to the nodes that have larger values of backoff counters. This introduces what is known as capture effect in Ethernet LANs resulting in ‘short-term unfairness’. The second part of the project deals with studying this short term unfairness in an Ethernet LAN.

The simulation was performed in ns2 (network simulator 2) which is a very commonly used discrete event simulator for computer networks. The simulation code was used to generate traces by passing different combinations of arguments to it like packet size and offered load. The generated trace files were parsed to generate data files for getting statistical results about the simulation and plotting the simulation results.

Some related works done in this field which we referred to were by Boggs et. Al [1] and Wang [2]. They also measure the Ethernet performance and give a general analytical overview of it.

II. SYSTEM MODEL FOR SIMULATION

This section describes the system model we used in our simulation. This includes the network topology, the nodes: how it interacts with the LAN, buffers at the sender nodes, the receiver etc. The system model can be best viewed by the following schematic diagram.

![System Model for Simulation](image)

Figure 1: The System Model for Simulation

As seen in figure 1, it shows the LAN wire where $n$ nodes are plugged into it. $(n-1)$ of them are sender nodes and one of them is a receiver node. The sender nodes inject packets into the network at a given offered load $\lambda$, as shown in the figure. Thus the total offered load to the network is $(n-1)\lambda$, which is also sometimes called the sent throughput. Every sender node maintains a buffer of size $B$ (counted in number of packets it can accommodate), which is nothing but a drop tail queue buffer. The Ethernet’s maximum capacity is 10Mbps as usual.

![A typical logical link in the simulation](image)

Figure 2: A typical logical link in the simulation

The above figure shows a typical logical link between a sender node and the receiver which is used in the simulation...
script. An UDP agent is attached, following Constant Bit Rate (CBR) traffic, the data rate being $\lambda$.

III. SIMULATION RESULTS

We simulated the network for 8 nodes where 7 of them were sender nodes and one was a receiver node. We used packets of size 64, 128, 256, 512, 1024 and 1500 bytes. Offered load was varied from 0.2 to 20 Mbps in steps of 0.2. A constant buffer size of 10 was used for all the experiments, except for the one testing the effect of buffer size on drop probability. The simulation time for each simulation run was adjusted such that the particular statistic stabilizes. We modified the ns2 script so that we can directly pass it command line arguments in the following form, which was done from a shell script for different combinations of input arguments:

```
$ns ethernetlan.tcl pktsize offeredload
```

This output was redirected to a trace file, which was parsed by a shell script to get presentable data. We used gnuplot to represent graphical data from the tabular data.

A. Throughput Characteristics

The following figure shows the Throughput characteristics with varying offered loads. We plot it for different values of packet sizes.

![Throughput vs. Offered Load](image)

Figure 3: Aggregated Throughput vs. Offered Load

As expected the throughput goes up with increasing offered loads and saturates after some point. Increasing offered load also increases the total number of bits transferred. Another important point to note is the effect of packet size. So, for a given buffer size and load; total number of bits that are successfully transmitted is higher for larger packet. As the packet size goes up it uses the channel more efficiently and thus channel utilization goes up resulting in enhanced throughput.

B. Delay Characteristics

The first type of delay we study is Network Delay. This latency is purely introduced by the network. We calculate it by the time elapsed between to dequeue the packet and it being received. Source to receiver delay consists of queuing delay, transmission delay and propagation delay. It increases with the increase of offered load. This is because the queuing delay increases as the source generates more packets. Also, larger packets encounter more transmission delay than smaller packets. So, the total delay increases with the increase of offered load and packet size.

![Network Delay vs. Offered Load](image)

Figure 4: Network Delay vs. Offered Load for packets of size 64, 512 and 1024 bytes

The second type of delay we worked upon is the Total Delay. This includes both the Network Delay and the Queuing Delay combined. This is calculated by the time difference between the packet enqueue and receive-time.

![Total Delay vs. Offered Load](image)

Figure 5: Total Delay vs. Offered Load for packets of size 64, 512 and 1024 bytes
The network delay increases with the increase of offered load, and then it goes down. It’s higher for larger packets; since larger packets encounter more transmission delay than smaller packets. It may be noted that network delay is much smaller than the total delay.

C. Drop Probability

This plot shows the dynamics of drop probability as a function of aggregated offered load. The load on the network is increased by increasing the individual offered load of a particular host keeping the number of hosts unchanged. As a consequence the aggregated offered load grows from 0.2Mbps to 20Mbs in steps of 0.2. We made the buffer size sufficiently large (500 packets) in order to ensure that buffer size is not contributing to drop rate of packets.

As we increased the total offered load the drop probability increased. This was more for smaller sized packets compared to larger sized packets. This is because smaller the size of the packets, more the number of packets given a constant data rate and more collisions. The drop probability decreases somewhat with increasing packet sizes.

D. Effect of Buffer Size

In this subsection we show the effect of buffer size on the drop probability. We kept the total offered load at 8Mbps and the packet size to be 512bytes. The buffer size was varied from 10 to 200 in steps of 10, and the corresponding drop probability was calculated.

As evident from the figure, as we increased the buffer size the drop probability decreased. Since, larger the size of the buffer more likely is that more number of packets get accommodated in the waiting queue, compared to smaller buffers where packets can get dropped easily as most often there won’t be room for holding packets. But the drop probability decreased steadily as we increased the buffer size and it became significantly low (around 0.2) at a value of 200 for the given set of network parameters. Thus buffer size plays an important role in determining the drop probability in a network.

E. Unfairness Issues

In order to demonstrate short term unfairness of Ethernet we take the approach of Molle [3]. As usual in our experiments we consider 7 nodes sending packets to a single destination node at 0.1Mbps. In figure 8 we can see the dependency of probability that the identity of the sender of a randomly chosen packet will be the Kth most recently seen source address on the network from Most Recently Used (MRU) stack depth. In case of a completely fair network the graph should be a horizontal line.

IV. CONCLUSION

Thus in brief we looked at the most fundamental characteristics of an Ethernet LAN and the various parameters they depend on. The characteristics include Throughput, Delay...
and Drop for different packet sizes and offered loads. Throughput increases with increasing offered loads and saturates after a given point. Increasing the offered load too much will decrease throughput due to huge number of collisions. Throughput is more for larger packets compared to smaller ones.

Next we studied delay characteristics – with increasing offered load the packet delay also increases. Two types of delay were studied: one was the Network Delay (purely the flight time of the packet) and the other was the net delay which includes queuing delays, link latency and network delays combined. Larger the size of the packet it faces more delay due to greater transmission time. Then we studied the drop probability of a packet with varying offered loads and buffer sizes. A large buffer size or a waiting queue decreases the drop probability. Secondly we dealt with presenting graphically the ‘Ethernet capture effect’ or the short term unfairness seen in Ethernet LAN.

V. APPENDIX

We devoted this section to briefly describe the simulation process we followed for this particular project work. The ns2 code was modified to take command line arguments for packet size and offered load. The shell script runs a nested for loop involving combinations of packet sizes and offered loads, and for every instance of the combination produces a trace called trace_pktsize_i_load_j and the awk script filters all of them, where the data is traced in another file. We use gnuplot program to graphically plot the data in the trace file. The following figure is a schematic diagram of our simulation work.

![Figure 9: A schematic representing our simulation](image)

- Some sample codes for the shell script, awk script and the modified ns2 code are provided at:

REFERENCES

