Design and Analysis of an Access Network based on PON Technology

Glen Kramer and Biswanath Mukherjee
Department of Computer Science
University of California, Davis, CA 95616, USA
Tel: 1.530.752.5129; Fax: 1.530.752.4767
E-mail: {*kramer, mukherjee*}*@cs.ucdavis.edu*

August 2000

Introduction

With the expansion of services offered over the Internet, the "last mile" bottleneck problems continue to exacerbate. Passive Optical Network (PON) is a technology viewed by many as an attractive solution to this problem [1,2].

There are several topologies suitable for the access network: tree, ring, or bus (Figure 1). All transmissions in a PON are performed between Optical Line Terminal (OLT) and Optical Network Units (ONU). Therefore, in the downstream direction (from OLT to ONUs), PON is a point-to-multipoint network, and in the upstream direction it is a multipoint-to-point network.

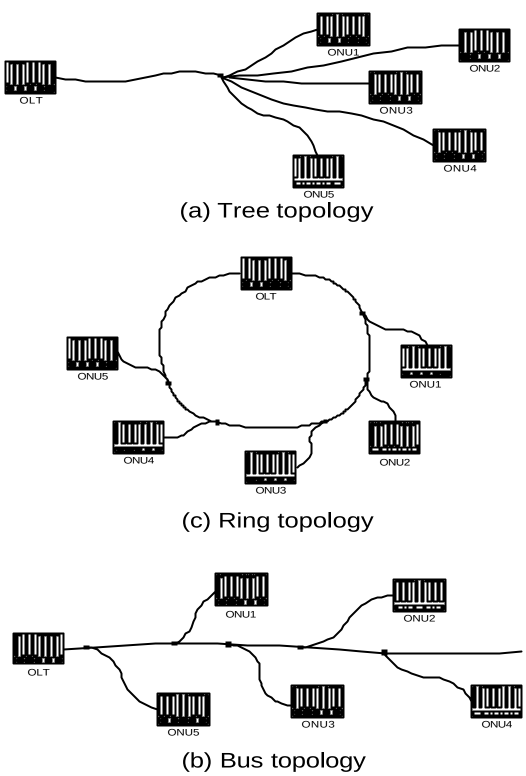

Figure 1. PON topologies.

In this study, we propose the design of a PON architecture that has an excellent performance-to-cost ratio. This architecture uses time-division multiplexing (TDM) approach to deliver data encapsulated in Ethernet packets. We consider a model with N ONUs. Every ONU is assigned a timeslot. All N timeslots together compose a frame. In all numerical examples presented in this study, the default value for N was chosen to be 16.

The simulation analysis was performed using Bellcore traces that exhibit the property of *self-similarity* [3]. Self-similar (or fractal) traffic has the same or similar degree of burstiness observed at wide range of time scales. Using self-similar traffic is extremely important as it provides realistic bounds on packets delay and queue occupancy.

Packet delay and queue size

Before we present our results, let us consider what are the constituents of the delay experienced by a packet. Packets arrive to the ONU at random times. Every packet has to wait for the next timeslot to be transmitted upstream. This delay is termed ***TDM delay***. TDM delay is the time interval between packet arrival and the beginning of the next timeslot.

Due to the bursty nature of network traffic, even at light network load, some timeslots may fill completely and still more packets may be waiting in the queue. Those packets will have to wait for later timeslots to be transmitted. This additional delay is called ***Burst delay***. Burst delay may span multiple frames.

In our simulations we found that both number and size of traffic bursts increases with the increase of network load. That leads to an exponential growth of packet delay and queue size as a function of load, thus making buffering extremely inefficient as a mean of preventing packet loss. The only solution is to increase available egress bandwidth (channel efficiency).

Bandwidth utilization

The egress bandwidth remains underutilized when timeslots cannot be filled completely. That happens if the next packet to be transmitted is larger than the remainder of the timeslot. Such packets will wait for the next timeslot, leaving an unused remainder at the end of the current timeslot.

We found that the expected value of the unused remainder can be obtained as

$$E(R) = \frac{1}{E(X)} \sum_{r=1}^{M-1} r \times [1 - F_X(r)] \qquad (1)$$

where X is a random variable representing packet sizes and M is the maximum packet size (1518 bytes for Ethernet traffic).

The amazing result here is that $E(R)$ does not depend on timeslot size. It only depends on the distribution of packet sizes. This agrees very well with our simulations.

It follows that the maximum utilization achieved by an ONU is

$$U = \frac{T - E(R)}{T} \tag{2}$$

where T is the timeslot size.

Then, it seems that in order to increase utilization we need to increase the timeslot size. However, that would result in increased packet delay. To find the optimal timeslot size we employ power function [4] defined as

$$P(t) = \frac{U(t)}{D(t)} \tag{3}$$

where $U(t)$ is the utilization as a function of timeslot size and $D(t)$ is the delay as a function of timeslot size.

Then, the maximum of the power function would give us the best timeslot size to achieve the best utilization-to-delay ratio. However, as we increase offered load, we notice that the maximum of power function shifts with exponential increments (see Figure 2).

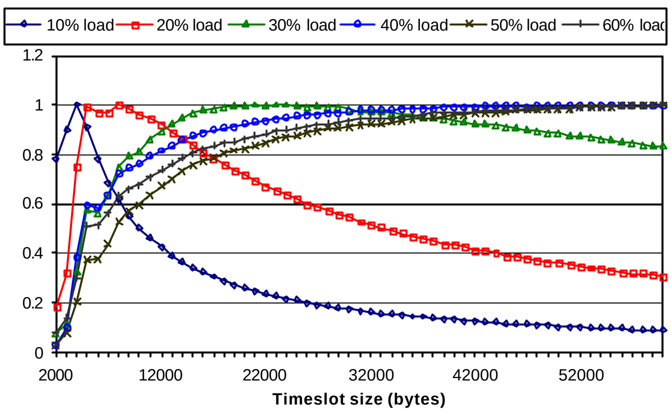

Figure 2. Normalized power function for different loads.

This is an interesting discovery. Not only does the Burst delay increase exponentially with increased load, but the maximum of the power function also shifts with exponential scale. This means that adjusting the timeslot size cannot be a solution, as it results in exponential increase of frame size, and hence exponential increase in TDM delay value. While the Burst delay can be avoided for high priority packets by using clever scheduling schemes, the TDM delay is a fundamental delay that affects all packets. Altering the timeslot size would mean that we are unable to provide any guarantees to the delay value or variation.

Thus, instead of increasing utilization by increasing timeslot size, we may attempt to increase utilization by decreasing the expected value of the unused remainder (refer to equation 2). To do that, we will consider the reordering of packets that are waiting in the buffer.

Packet reordering (scheduling)

The reordering of packets waiting in the buffer is a variation of the bin-packing problem. Different flavors of the algorithm may be used: first fit, best fit, prediction, etc. However, as it turns out, packet reordering based solely on packet sizes is not such a good approach. To understand the problem, we need to look at the effects of packets reordering from the perspective of TCP/IP payload carried by Ethernet packets. Even though TCP will restore the proper sequence of packets, an excessive reordering may have the following consequences:

1) According to fast retransmission protocol, the TCP receiver will send an immediate ACK for any out-of-order packet, whereas for in-order packets, it may generate a cumulative acknowledgement (typically for every other packet) [5]. This will lead to more unnecessary packets being placed in the network.
2) Second, and more important, packet reordering in ONU may result in a situation where 4 or more later packets are being transmitted before an earlier packet. This would generate 4 ACKs (3 duplicate ACKs) for the earlier packet. That will trigger packet retransmission and reduction of the TCP's congestion window size (the *cwnd* parameter) (refer to the Fast Retransmission protocol in [5] or elsewhere).

Even if special care is taken at the ONU to limit out-of-order packets to only 1 or 2, the network core may contribute additional reordering. While true reordering typically generates less than 3 duplicate ACKs and is ignored by the TCP sender, together with reordering introduced by the ONU, the number of duplicate ACKs may often exceed 3, thus forcing the sender to retransmit a packet. As a result, the overall throughput of user's data may decrease.

The solution is to reorder only those packets that belong to different connections. Thus, we propose an algorithm that will put smaller packets ahead of larger packets only if they belong to different connections, i.e., they have different senders, or different receivers, or belong to different applications.

This connection-oriented first-fit algorithm improves utilization by 1.5% to 4% (depending on timeslot size). While this algorithm may be too computationally expensive weighting against its benefits, it may be implemented as part of QoS scheduling at the very low additional cost.

Bibliography

[1] G. Pesavento and M. Kelsey, "PONs for the broadband local loop," Lightwave, PennWell, vol. 16, no. 10, pp. 68 – 74, September 1999.
[2] B. Lung, "PON architecture 'futureproofs' FTTH", Lightwave, PennWell, vol. 16, no. 10, pp. 104 – 107, September 1999.
[3] W. Leland, M. Taqqu, W. Willinger, and D. V. Wilson, "On the self-Similar Nature of Ethernet Traffic (extended version)," *IEEE/ACM Transcations on Networking*, vol. 2, pp. 1-15February 1994.
[4] L. Kleinrock, "On Flow Control in Computer Networks," Conference Record, *Proceedings of the International Conference on Communications*, vol. II, pp. 27.2.1 - 27.2.5, Toronto, Ontario, June 1978.
[5] W. R. Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley Publishing Co., Reading, MA, 1994.