# Design and Analysis of an Access Network based on PON Technology

***UC Davis Student Workshop on Computing '2000***

Glen Kramer

kramer@cs.ucdavis.edu

September 30, 2000

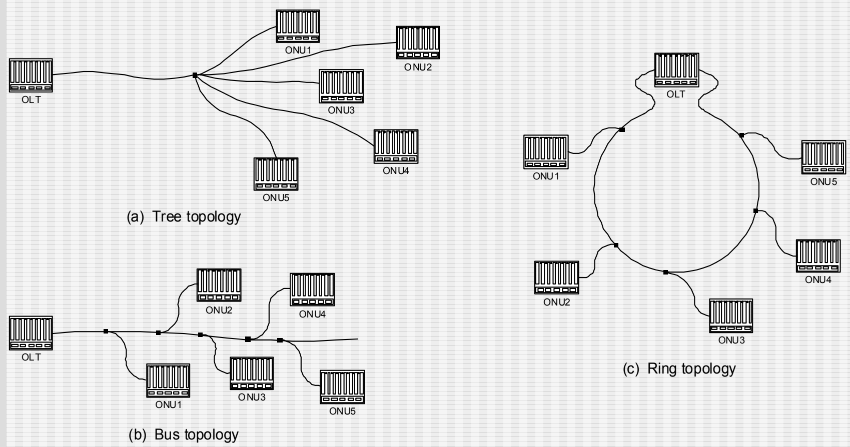
# What is Passive Optical Network?

Passive Optical Networks (PON) are point-to-multipoint optical networks with no active elements in the signals' path from source to destination.

Advantages of PON:

1. PON allows longer distances between CO and customer: 20 km for PON vs. 5.5 km for DSL.
2. PON Provides higher bandwidth.
3. Allows downstream video broadcasting.
4. Eliminates electronic devices in the middle of the network.
5. Allows easy upgrades to higher bit rates or additional wavelengths.

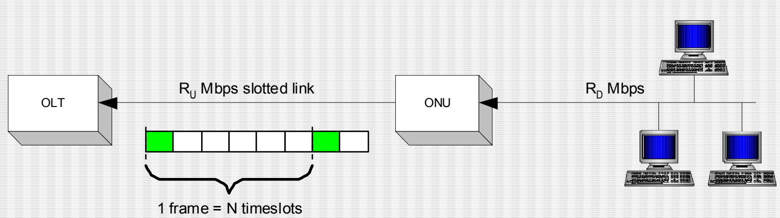
# PON topologies






# Simulation Model

- Most efficient architecture:
  1. Every ONU assigned a timeslot (TDM channel)
  2. Data encapsulated in Ethernet frame.



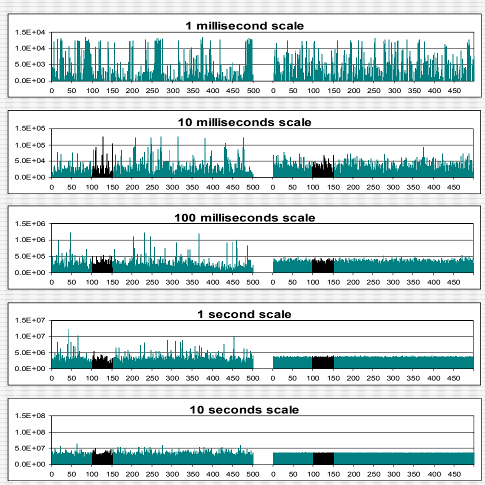
## Goals of this study

- What is the average packet delay.
- What is the average queue size.
- What link utilization we can achieve (link efficiency).



## What is self-similar traffic?

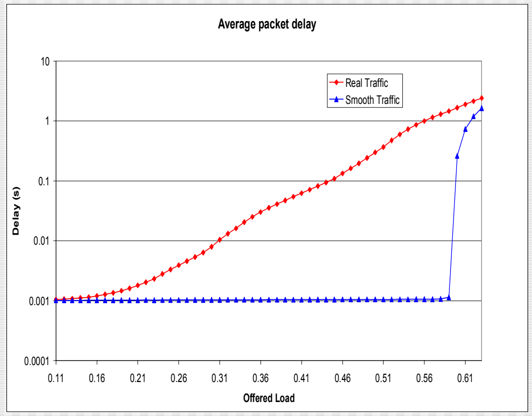
Self-Similar (LRD) traffic




Poisson Arrivals

# Average Packet Delay

- Frame time F = 2 ms
- N = 16
- Unlimited buffer size
- No packets dropped
- Maximum upstream bandwidth 1Gb/s /N = 62.5 Mb/s

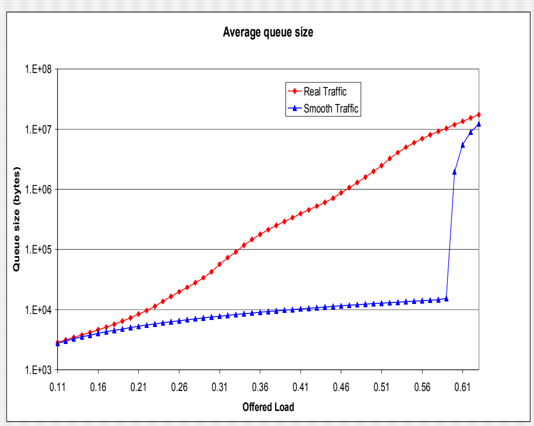





# Average Queue Size

Timeslot size = 15625 bytes. This value is defined by the frame time, number of ONUs, and line rate.

$$\begin{aligned} Timeslot(bytes) &= \\ &= \frac{Frame(s)}{N} \times Line\_Rate \\ &= \frac{2\times10^{-3}s}{16} \times 10^9 \frac{bit}{s} \times \frac{1}{8}\frac{byte}{bit} \\ &= 15625(bytes) \end{aligned}$$



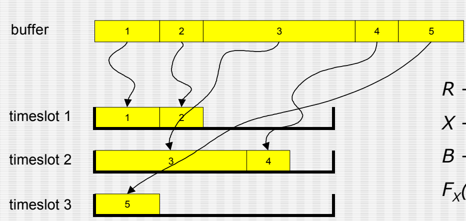
# Simulation Results

- No traffic smoothing possible
  (burst size distribution is heavy-tailed).
- Large buffers will not prevent congestion or packet loss.
- Higher egress bandwidth is required to prevent congestion.



# Why is link efficiency less than 100%?

- Ethernet Packets cannot be fragmented.
- If the next packet to be transmitted is larger than the remainder of timeslot, the packet will wait for the next timeslot =>the timeslot will be transmitted with an unused remainder at the end.




*R* – r.v. representing unused remainder

*X* – r.v. representing packet size

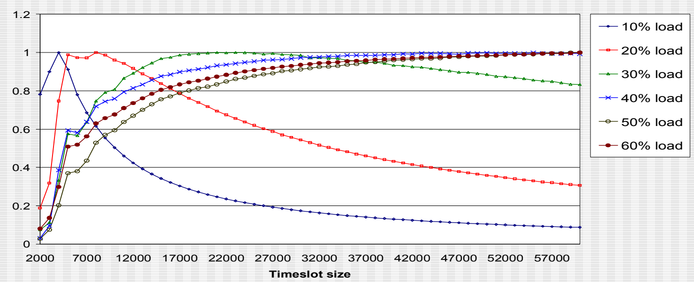
*B* – maximum packet size

$F_X(x)$ – CDF for X

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

## First attempt to improve link utilization

- Utilization *U = 1 – E(R) / Timeslot*
- Increasing timeslot will improve utilization, but will increase packet delay.
- Solution: use power function P to find the optimal timeslot size (*P = Utilization/Delay*).





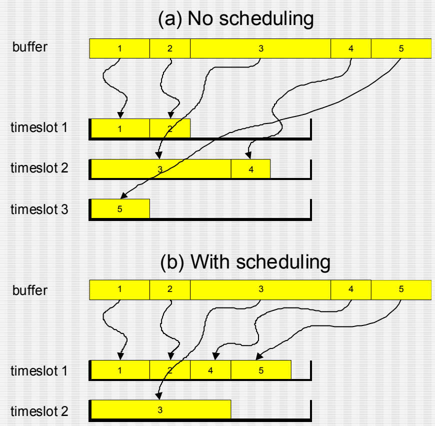

## Why timeslot adjustment won’t work

- Linear increase in offered load requires exponential increase in timeslot size.
- Increased timeslot size will increase timeslot period => will increase packet delay.
- Timeslot adjustment should be based on traffic load. However, due to burstiness of traffic at every timescale, no load prediction is possible based on previous load.

## Second attempt to improve link utilization

*U = 1 – E(R)/Timeslot*

Try to reduce *E(R)* by scheduling (reordering) packets that are already in the buffer.

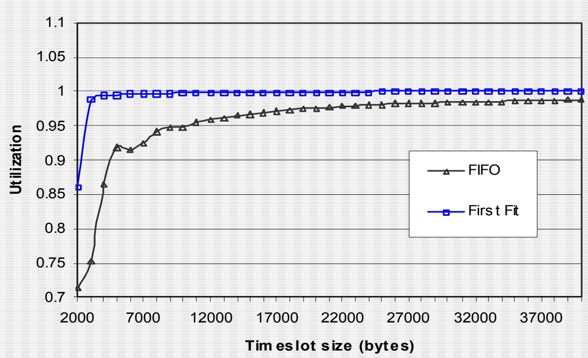





## Result of packet scheduling

Assuming unlimited supply of packets, how well can we fill the timeslots?

**FIFO** – no packet reordering.
**First Fit** – search the buffer for the next packet that fits.



## Scheduling may be bad

1. Receiver will send an immediate ACK for any out-of-order packet => unnecessary packets placed in the network.

2. If 3 or more packets are sent out-of order, Sender will receive 3 or more duplicate ACKs. Then, Sender will retransmit the packet and reduce its congestion window size ( the *cwnd* parameter).



## Smart Scheduling

- Traffic is an aggregate of multiple connections (flows).
- Connection may be identified by combination of source and destination IP addresses and port numbers.
- Reorder packets that belong to different connections (flows).

# Smart scheduling algorithm

Let $Q$ be the queue of packets $q_1, q_2,.. q_n$ waiting in ONU
$C(q_i)$ – connection id of packet $q_i$
$P$ – set containing Ids of packets that were postponed
$R$ – slot remainder

Repeat for every timeslot
{
  $i = 1$
  $P \in \varnothing$ (Clear the set P)
  $R = |timeslot|$
  While $i \le n$ and $R \ge \min$
  {
    If $q_i \le R$ then (packet fits into timeslot)
    {
      if $C(q_i) \notin P$ (i.e. packets from this connection were not postponed yet)
      {
        send $q_i$.
        $R = R - |q_i|$
      }
    }
    else (packet doesn't fit into timeslot)
      $P = P \cup C(q_i)$ (add connection id to P)
    $i = i + 1$
  }
}



# Conclusion

- Large data bursts were present at a very low average load => packet loss could not be prevented. Having larger buffers will slightly reduce the congestion, but will increase the delay.
- Bandwidth is lost due to unused remainders at the end of timeslots. Solution (a) increase timeslot or (b) reduce the remainder (by reordering packets).

  a) Larger timeslots => larger delay.
  b) Scheduling reduces the remainder, but requires more CPU power.