MEDIUM ACCESS CONTROL IN LIGHT TRAIL AND LIGHT BUS NETWORKS

Srivatsan Balasubramanian, Ahmed E. Kamal, Arun K. Somani
Department of Electrical & Computer Engineering
Iowa State University, Ames, Iowa 50011, USA
{vatsan, kamal, arun}@iastate.edu

Abstract The internet transport infrastructure is evolving towards a model of wire speed IP routers interconnected by intelligent optical networks. Recently, a new architecture called the Light Trail has been proposed that provides a novel and amenable control and management solution to address IP-centric communication issues at the optical layer. This paper reviews the concept of Light Trails and proposes a variant of it called the Light Bus. The focus of this paper is to design a simple medium access control protocol for the Light Trail and the Light Bus networks. We present an exact analytical model to estimate the queuing delay distribution characteristics of the protocols for a small network. For bigger networks, we simulate the protocols and analyze their behavior.

Keywords: Light Trail, Light Bus, CSMA, Queueing Theory.

1. Introduction

The unprecedented growth of internet traffic and rapid advancements in the optical transport technologies has led to research and development in the optical Internet. In such a dynamic environment, a network architecture that can enable high speed provisioning, accommodate multi-granularity traffic and support high data rates will be the key feature of the next-generation Internet.

Among all the optical networking paradigms, wavelength routed networks have been the subject of intensive research for the past few years. By setting up a dedicated lightpath between node pairs, signals can be transported transparently at different rates, and in different protocol formats. However, it is increasingly becoming evident that the individual traffic streams that these circuit switched networks will carry are likely to have bandwidth requirements less than that of a wavelength.

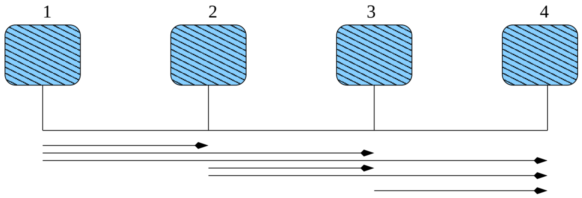


Figure 1. A Light Trail with 4 nodes allows up to 6 unidirectional connections with the wavelength being shared in time.

A wavelength is still a scarce resource but the majority of the resource allocation in the wavelength routing approach is coarse grained. Hence the wavelength capacities may be severely underutilized and result in the increased usage of wavelengths thereby leading to high network costs.

An alternative to circuit switching is optical packet switching [7] which achieves efficient network utilization since it leads to high statistical multiplexing gains and is amenable for traffic engineering. In optical packet switching, the payload remains in the optical form while the header may be processed optically or electronically. However, the requirement of synchronization and the lack of large and fast optical memory and switching units prevents implementation of router architectures in optics that are possible in electronics.

Optical burst switching (OBS) [2, 4] was proposed as a switching paradigm that allows bursts of data to be switched through the network all-optically. The key idea in optical burst switching is to decouple the header from the payload. The header is transmitted ahead of the burst to configure all the crossconnects in the route and the data burst follows the header after an offset time. The offset time accounts for the header to be processed in each node and the switches to be configured. With the present optical switching technology, the ratio of configuration time of optical crossconnects to burst duration is very high and since this control overhead is encountered for every burst, the network utilization can be very low.

IP is gaining popularity in becoming the convergence layer for the global and ubiquitous internet. IP traffic is known for its burstiness and high variability and as discussed above, the existing architectures are either inherently unsuitable or too expensive and inefficient to handle such traffic. Recently, a new architecture called Light Trails [1] has been proposed to transport IP over WDM that offers the advantages of all the existing architectures but still avoids the limitations of the same.

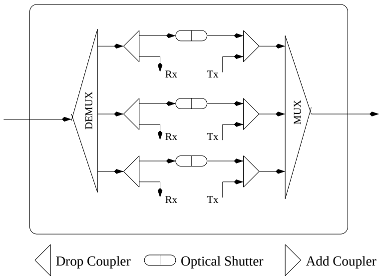


Figure 2. Light Trail node architecture

By utilizing mature optical components in a novel way, eliminating the requirements of high speed optical switches and enabling support for sub-wavelength provisioning and efficient multicasting, Light Trails present a good potential for successful mass deployment in the access and metro arenas. This paper reviews the concept of Light Trails and a variant of it called the Light Bus and develops and analyzes medium access control protocols for both these architectures.

2. Light Trail Architecture

A Light Trail is similar to a lightpath in that it requires the establishment of an all-optical path passing through several intermediate nodes. The key difference is that any intermediate node in a Light Trail can also transmit and receive data on the same channel unlike in the lightpath where only the end nodes can exchange information. Consider a 4 node Light Trail that has been designed such that the sum of the traffic sourced by all the nodes is less than the capacity of a wavelength. Each of the nodes is allowed to transmit data to any of their respective downstream nodes without the need for switch reconfiguration. In a unidirectional Light Trail with N nodes, up to $\binom{N}{2}$ connections are possible. Figure 1 illustrates a Light Trail with nodes $\{1, 2, 3, 4\}$ and all possible traffic streams that can be accommodated in the trail. A total of up to six s-d pair can be supported - (1,2), (1,3), (1,4), (2,3), (2,4) and (3,4). The wavelength is shared in time by all the nodes and the medium access is arbitrated by a control protocol that prevents collisions among the nodes that try to transmit data simultaneously. When a node transmits a packet on a wavelength, a portion of the power in the wavelength is tapped by every node through which the signal passes, and the node decides whether to accept or ignore the packets based on electronic processing of the packet. This mechanism inherently allows

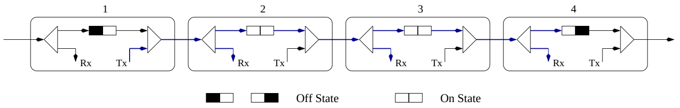


Figure 3. Data transfer from node 1 to node 4 in a Light Trail. The optical shutter is in the off state on node 1 and node 4 and in the on state on node 2 and node 3. Data from upstream preempts any transmission on downstream nodes.

multicasting without the need for special hardware support. An out-of-band control channel carrying information pertaining to channel set up, tear down and dimensioning of Light Trails is dropped and processed on every node in the trail. The exclusion of high speed switching coupled with inherent multicasting capabilities and flexible and fast provisioning for diverse granularity traffic makes the Light Trail architecture superior to the circuit, packet and burst switched architectures.

The node and the switch architecture of the Light Trail are shown in Figure 2. Wavelengths from an input link are filtered and sent to their respective light trail switches. A portion of the signal is tapped at the receiver and the remaining signal is passed through the ON/OFF light switch (typically an acousto-optic tunable filter). For a node that is the start or the end node, the ON/OFF switch is in the OFF position thereby blocking the signals from propagating further. For other intermediate nodes, the switch is in the ON position, thereby allowing the signals to pass through the node. The configurations of the start, intermediate and the end nodes are shown in Figure 3. These switches are set up in the required positions during the Light Trail design phase and are not reconfigured for every packet. This leads to an excellent provisioning time within a Light Trail as compared with other protocols like OBS.

A simple medium access control protocol which belongs to the CSMA class of protocols was proposed for the Light Trails in [1], which gives higher priorities to upstream nodes. When a node (say, A) wants to transmit data, it senses the channel to determine if it is being used by an upstream node. If the channel is free, A immediately sends a beacon signal indicating that it has a packet to transmit. If the channel is busy, A waits until the channel is free and then sends the beacon signal. At some fixed offset time (also called the guard band)after the beacon is sent, A starts transmitting its data. A continues to sense the channel while its packet is being transmitted. In the middle of the transmission, A may hear a beacon signal from an upstream node (say, B) which has data to transmit and once A senses this, it abruptly stops its transmission and lets B's packet to pass through safely. The packet

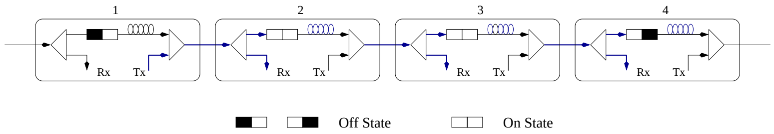


Figure 4. Data transfer from node 1 to node 4 in a Light Bus. The optical shutter is in the off state on node 1 and node 4 and in the on state on node 2 and node 3. The delay lines buffer upstream data until the data downstream completes transmission.

that was prematurely truncated by A is discarded by the receivers since it will fail the link level error checks. Thus, by sending a beacon signal before actually transmitting a packet and by always giving priority to the upstream nodes, the protocol successfully resolves medium access contentions.

In Light Trail networks, the key design parameter that affects protocol performance is the guard band gap. The guard band should be set large enough to allow the receiver to sense the beacon signal and pull out of transmission and small enough avoid to avoid significant throughput degradation due to excessive overhead. In general, the receivers are moderately fast and the carrier signal from the upstream node can be sensed in a few nanoseconds. The laser is also very fast and the external modulator can be turned on and off in sub-nanosecond time regimes when working at 10 Gbps, and so pulling out of transmission in the case of collisions does not take too long. But, when the decision to stop is made by the receiver and is propagated to the transmitter, the feedback control will happen through a microcontroller and hence the delay would be large and we expect it to be at least between 50 ns and 100 ns. This overhead has to be incurred by every packet on every node except on the penultimate node which does not have any other downstream node to collide with.

2.1 Light Bus Architecture

We propose a new architecture called the Light Bus which is a variant of the Light Trail architecture. The key difference is that the Light Bus switch architecture includes a fiber delay loop on every node. The delay accounts for the time needed to transmit a maximum size packet. This delay time allows the transmitter to complete its current transmission when a transmission from an upstream node is detected on the lightbus. The lightbus medium access control protocol works as follows.

When a node has data to transmit, it first checks for activity on the delay line. If the delay line is free, the node goes ahead and completes its transmission. If the delay line is busy, the node waits till the delay line becomes free and then starts transmission. In the middle of a transmission, if a node upstream starts sending data, it is buffered in the delay line of the downstream node until the current transmission on the node is completed. Appendix B gives a pseudocode for both the Light Trail and the Light Bus protocols.

The key idea in the Light Bus architecture is to design a more amenable and efficient control structure on both the sender and the receiver nodes. The transmitter does not have to worry about dropping, suspending and retransmitting packets and the receiver does not have to deal with runt packets. Unlike the Light Trail protocol, the Light Bus approach allows for all the nodes in the bus to send out data simultaneously (under certain conditions) and does not lead to wasted transmissions that result in increased power consumption. Besides, a guard band on every packet is required by the Light Trail protocol whose duration depends on the rise and fall times of the transponders and the speed of the control electronics. This additional overhead incurred for every packet is avoided in the Light Bus protocol. However, the presence of the fiber loops impose extra queueing and propagation delays. Thus, the central idea behind the Light Bus model is to trade-off performance for simplicity and efficient control.

The Light Bus and Light Trail medium access protocols are modeled in section 3 and the simulation results are analyzed in section 4. Section 5 presents a discussion on observed behavior of the protocols based on the analytical model and suggests some performance enhancements to the Light Bus protocol. Finally, we present some conclusions based on our current research and outline future directions.

3. Analytical Model

In this section, we present the exact model of the delay characteristics of a 3-node Light Trail and the 3-node Light Bus protocols. It becomes difficult to extend the model to more than three nodes as the traffic characteristics changes after the second node. However, an approximate model that aggregates the traffic coming from upstream nodes and representing it by a poisson process is feasible and will be identical to the model presented in this section. To facilitate our presentation, the following notations will be used for packets on each queue.

W_2 - Virtual service time of a node 2 packet that arrives to a nonempty queue.

W_2^0 - Exceptional service time of a node 2 packet that arrives to an empty queue.

C_2 - Completion time of a node 2 packet.

B_i - Duration of the busy period.

E - Duration of the extended busy period.

S_i - Duration of silence.

R_E - Residual life time of E seen by an incoming packet.

$\phi_E(z)$ - Moment Generating Function (MGF) of E.

$\phi_S(z)$ - MGF of S.

$\phi_{R_E}(z)$ - MGF of R_E.

$\phi_T(z)$ - MGF of waiting and service time of a node 2 packet.

$\bar{B}, \bar{E}, \bar{R_E}$ - Mean values of the busy period, extended busy period and the residue of the extended busy period.

b_1, b_2 - Service time of a node 1 and node 2 packet.

$\bar{b_1}, \bar{b_2}$ - Mean values of node 1 and node 2 packet service time.

λ_1, λ_2 - Packet arrival rates at node 1 and node 2.

ρ_1, ρ_2 - Utilization factor of queues on node 1 and node 2.

D - Maximum service time of a packet.

Note that the definitions of some of these terms are explained in the next subsection.

3.1 The Light Bus model

We analyze the delay distribution characteristics of a three-node Light Bus which starts from node 1, passes through node 2 and ends at node 3. The third node of the Light Bus is not involved in transmission

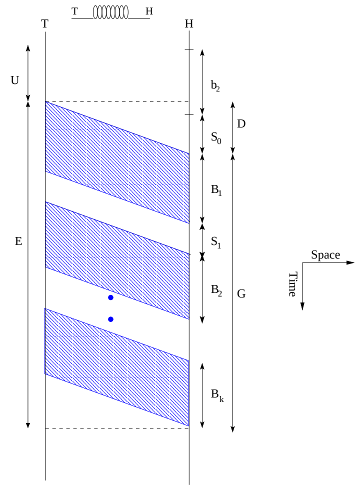


Figure 5. Space-Time diagram for the Light Bus model. The two vertical lines T and H refer to the two end points of the delay line on node 2. T denotes the tail of the delay line where the packet enters and H denotes the head of the delay line where the packet exits.

of data and hence only the first two nodes are considered. We ignore the propagation delay since the Poisson arrival process is stationary and hence unaffected by time shifts. The behavior of queue 1 can be modeled as an M/G/1 system since the service time of the packets of queue1 is independent of the arrivals on other queues. The Laplace transform of the waiting time PDF is given by the Pollaczek-Khinchin transform equation [3] as

$$\phi_{W_1}(z) = \frac{z(1-\rho_1)}{z - \lambda_1 + \lambda_1 \phi_B(z)} \tag{1}$$

The mean queuing delay of the packets on node 1 is given by

$$W_1 = -\phi'_{W_1}(0) = \lambda_1 \bar{b_1^2}/2(1-\rho_1) \tag{2}$$

To understand the behavior of the second queue, refer to the space-time diagram in Figure 5. The space-time diagram depicts a temporal sequence of transmission activity at the tail end and the head end of the delay line on node 2. A transmission of a node 1 packet marks the beginning of a busy period, that is defined to begin with the arrival of a packet to an idle channel and to end when the channel next becomes idle. In the time it takes for the trailer of the last packet of the busy period to clear the head end of the delay line (which is D), if there is another arrival on node 1, the second busy period starts. This cycle repeats itself until the time when the last packet of the current busy period clears the head end of the delay line without a new arrival starting off another busy period. We refer to the time from the beginning of the first busy period until the end of last busy period as the "extended" busy period. The extended busy period, denoted by E in Figure 5, is marked by a series of active periods (denoted by B_i) interspersed with periods of silence (denoted by $S_i, 0 < S_i < D$). Queue 2 can be considered to be a variant of an M/G/1 system where the first message of each busy period sees an exceptional service time of W_2^0 and the other packets see a virtual (or extended) service time of W_2 as explained below.

Consider a packet that arrives to an empty queue at node 2. Since, Poisson Arrivals See Time Averages (PASTA), with a probability q, the transmission line is inactive and hence it is immediately serviced, and with a probability $1 - q$, the transmission line is active and the packet waits for residual time of E, called R_E, before it is serviced. The service time of the queue 2 packet is denoted by b_2. During the service time of the queue 2 packet, a busy period B_1 may start at queue 1, which is seen at head end of the delay line after a silence period of S_0. The period between the start of the service time b_2 and the start of busy period B_1 (as seen at the tail end of the delay line) is denoted by U. The first busy period may be followed by a series of B_i and S_i periods. Therefore, the total time elapsed after head of the queue 2 gets serviced and before the next packet (if any) on queue 2 can go into service is given by S_0+ G, where

$$G = \sum_{i=1}^{k-1} (S_i + B_i) + B_k$$

where k is the last busy period, as seen on the head end of the delay line, during which an arrival does not happen on queue 1.

The exceptional service time of the first packet is

$$W_2^0 = q(b_2 + S_0 + G) + (1 - q)(R_E + b_2 + S_0 + G) \tag{3}$$

The virtual service time of the other packets begin when the queue 2 packet begins transmission and ends when the packet is serviced and the system is clear of all queue 1 packets. So, the virtual service time is given by

$$W_2 = b_2 + S_0 + G \tag{4}$$

and their corresponding MGFs are given by

$$\phi_{W_2}(z) = \phi_{b_2}(\lambda_1 + z) + \phi_{b_2+S_0}(z)\phi_B(z)\sum_{n=0}^{\infty}\phi_{B+S}^n(z)(1-e^{-\lambda_1 D})^n e^{-\lambda_1 D}$$

$$= \phi_{b_2}(\lambda_1 + z) + \frac{\phi_{b_2+s_0}(z)\phi_B(z)e^{-\lambda_1 D}}{1-(1-e^{-\lambda_1 D})\phi_B(z)\phi_S(z)} \tag{5}$$

since B_i and S_i are independent and

$$\phi_{W_2^0}(z) = \phi_{W_2}(z)(q + (1-q)\phi_{R_E}(z))$$

$$= [\phi_{b_2}(\lambda_1 + z) + \frac{\phi_{b_2+S_0}(z)\phi_B(z)e^{-\lambda_1 D}}{1-(1-e^{-\lambda_1 D})\phi_B(z)\phi_S(z)}][q + (1-q)\phi_{R_E}(z)] \tag{6}$$

The MGF of waiting and service time of packets in queue 2,$\phi_T(z)$, can be obtained if $\phi_{W_2}(z)$ and $\phi_{W_2^0}(z)$ are known and is given by [8]to be,

$$\phi_T(z) = \frac{[1-\rho_2][\lambda_2\phi_{W_2}(z) - (\lambda_2 - z)\phi_{W_2^0}(z)]}{[1 + \lambda_2\bar{W}_2^0 - \rho_2][z - \lambda_2 + \lambda_2\phi_{W_2}(z)]} \tag{7}$$

The derivations of expressions for $\phi_B(z), \phi_{R_E}(z), q, \phi_{b_2+S_0}(z), \phi_{S_i}(z)$, and $\phi_{b_2}(\lambda_1 + z)$ are presented in Appendix A.

3.2 The Light Trail model

In the Light Trail approach, any upstream node can interrupt a downstream node and the downstream node retransmits the failed packet after the upstream node completes its transmission. The retransmission can also be interrupted and hence, the downstream node may have to try repeatedly until its packet is successfully transmitted. As in the case of a Light Bus, the first node can be treated as a simple M/G/1 queue and the Pollaczek-Khinchin transform equation can be applied to obtain the queueing delay of the node 1 packets. The second node can be modelled as a preemptive priority queue with restart-identical discipline. This is a well known problem and a solution for this is discussed in [5]. The completion time C_2 of a queue 2 packet begins when the system is clear of queue 1 packets and ends when service time b_2 has been completed and the system is clear of queue 1 packets. Notice that C_2 includes all interruptions by queue 1 packets, as well as all the service time of the queue 2 packet which must be repeated. The beacon signal transmission time and guard band gap that are required by the Light Trail protocol are accounted for by adding this time to the service time of the packet.

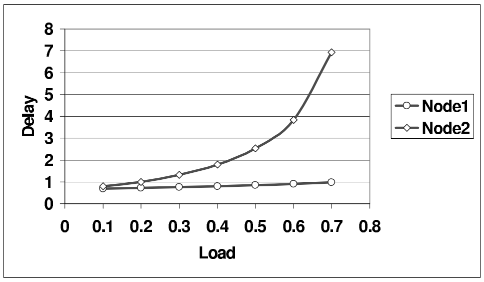


Figure 6. Queueing and service time delay Vs Load for the Light Bus protocol based on the analytical model.

The first and the second moments of the completion times are found to be [5]

$$E(C_2) = [\frac{1}{\lambda_1} + E(b_1)]E[e^{\lambda_1 b_2} - 1] \tag{8}$$

$$E(C_2^2) = 2[\frac{1}{\lambda_1} + E(b_1)]^2 E[(e^{\lambda_1 b_2} - 1)^2][E(b_1^2) + \frac{2E(b_1)}{\lambda_1} + \frac{2}{\lambda_1^2}]\times$$
$$[E(e^{\lambda_1 b_2}) - 1] - 2[E(b_1) + \frac{1}{\lambda_1}]\int_0^\infty x_2 e^{\lambda_1 x_2} b_2(x_2)dx_2 \tag{9}$$

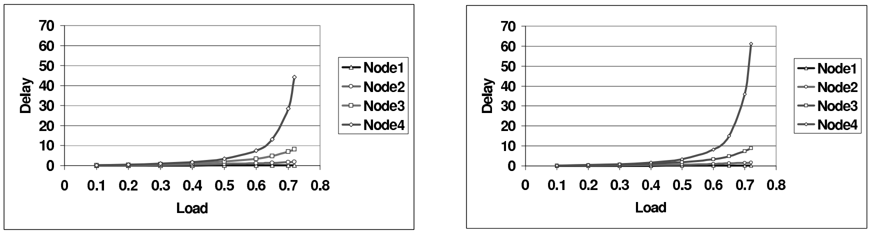
The average waiting time of the packets is given by [6]

$$\bar{W}_2 = \frac{\lambda_2 E(C_2^2)}{2[1 - \lambda_2 E[C_2]]} + \rho_1 \bar{R}_E \tag{10}$$

It is to be noted that no product form expression for the MGF of C_2 can be obtained due to the fact that the restart-identical model of service implies a non-memory less process.

4. Simulation Results

Performance analysis of the protocols for different traffic loads are done using discrete event simulation techniques. Infinite buffers are assumed to be present on all the nodes and the arrival process is Poisson distributed. The simulations are done for 10 Gbps systems and the results are obtained after transmitting 10 million packets. We assume the guard band required by the Light Trail to be 75 ns. In all the graphs, the x-axis plots the total offered load expressed as a fraction of the capacity of the wavelength and the y-axis plots the delay normalized to the time



(a) (b)

Figure 7. Queueing Delay Vs Load in (a) Light Bus (b) Light Trail when packet sizes are uniformly distributed between 500 and 1500 bytes.

taken to transmit a maximum sized packet. The traffic distribution is computed as follows.

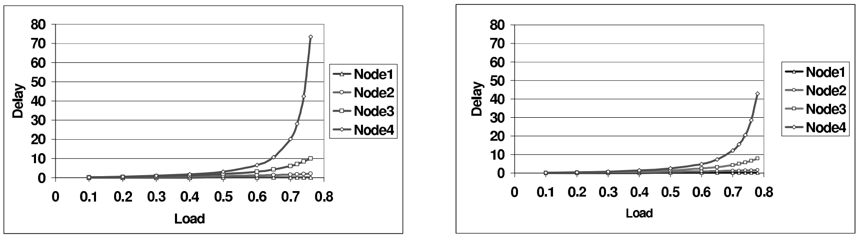
Consider a Light Bus or Light Trail of length N. The convener node is Node 1 and the end node is Node N. The traffic is assumed to be uniform for all s-d pairs. If T is the total traffic carried by the Light Bus, t_i is the traffic between any two s-d pairs, and C is the total capacity of the lightbus, then the traffic sourced by a node i is given by

$$t_i = \frac{(N-i)T}{N(N-1)/2} \tag{11}$$

and the system has been designed such that $\sum_{i=1}^{N-1} t_i \leq C$

In Figure 6, we plot the delay (sum of waiting time and service time) characteristics for a 3-node system based on the values obtained from our analytical model discussed in section 3.1. The packet sizes are uniformly distributed between 500 bytes and 1500 bytes. The traffic offered by nodes 1 and 2 are in the ratio 2:1 respectively. We see that the delay of the node 2 packets is very high and increases exponentially with increase in load since the node 1 packets are always given higher priority leading to starvation of node 2 packets. We performed simulations for this three node case and found our simulation results (not shown here) to match well with our analytical model.

We also performed simulations for bigger systems that are not covered by the model. We considered five-node Light Trails and Light Buses since the study in [1] observes that the average expected length of the Light Trails is five. The limit in the number of nodes occurs because of power budget constraints. The input power should be large enough to allow tapping of the signal in the intermediate nodes and yet small enough to prevent undesired non-linear interactions in the fiber.



(a) (b)

Figure 8. Queueing Delay Vs Load in (a) Light Bus (b) Light Trail when aggregated packet sizes are uniformly distributed between 16KB and 32 KB.

By Equation 11, for a five node system, traffic sourced by nodes 1 through 4 are in the ratio 4:3:2:1 respectively. Our general observation is that both the Light Trail and the Light Bus perform well until about loads of 0.7 C after which the delay increases exponentially. The delay encountered by a packet on any node is less than the delay for a packet on nodes that are futher downstream. For results shown in Figure 7, the packet sizes are uniformly distributed between 500 and 1500 bytes. When the packet sizes are small, the guard band required by the Light Trail protocol is of the order of the packet transmission time and the Light Bus protocol is able to outperform the Light Trail protocol by avoiding this overhead. In Figure 8, packet aggregation is assumed and the sizes are uniformly distributed between 16 KB and 32 KB. In this case, the guard band overhead is negligible when compared with the aggregate packet transmission time and hence the Light Trail protocol does better. It is observed that, in both the cases, the queueing delay is highest for the $N-1^{th}$ node while the delays are negligible for all the other nodes. This is understandable since the upstream nodes are always given higher priority over the downstream nodes and the penultimate node suffers the most.

5. Discussion and future work

The Light Trail approach yields a slightly better performance than the Light Bus approach for bigger packet sizes since it transmits a packet at queue 2 at the end of the busy period from queue 1, which has a probability of success equal to $\phi_{b_2}(\lambda_1)$. Since the Light Bus approach always forces queue 2 packets to defer their transmission if they observe a packet coming downstream from queue 1 during D time units, then even short packets, which can be otherwise accommodated under the Light

Trail approach, will have to wait under the Light Bus approach. That is, the probability of success under the Light Bus approach is $e^{-\lambda_1 D}$, which is always less than $\phi_{b_2}(\lambda_1)$ because $0 < b_2 \leq D$. We propose to improve upon the performance of the Light Bus protocol by implementing the following ideas.

The introduction of the delay line in the Light Bus approach actually gives an advantage, which is not present in the Light Trail approach, that can be capitalized. Namely, we know what is going to happen during the next D time units. We therefore, propose an approach which will yield a throughput, which is as good as, or even better than that produced by the Light Trail approach. When a node has data to transmit, if the delay line is empty, the packet at the head of the queue is transmitted. If the delay line is not empty, one of the following is done: if the gap on the delay line would allow the transmission of the packet at the head of queue 2, then it is transmitted. Otherwise, a fragment of the packet is transmitted, which can be easily done, and is supported by both IPv4 and IPv6. In order to reduce the fragmentation processing overhead and to streamline the fragmentation process, predefined fragment sizes can be used, like 1/4 MTU, 1/2 MTU and 3/4 MTU.

The processing overhead due to fragmentation depends on which protocol is being used. If the standard IPv4 protocol (or even IPv6) is used, then the overhead is basically computing the checksum and setting the values for the More, Offset and the Length fields. The checksum for all the bits in the header except for these fields are precomputed. Once the fragment size is decided, which can be 1/4, 1/2, 3/4 or 1 MTU, the offset field is set to this word, the More bit is updated and these bits are added to the checksum to generate the final checksum. Therefore, number of operations required for fragmentation are minimal and the introduction of fragments will not lead to a significant overhead. Also, unlike in a conventional packet switched network, packet reordering and packet losses are not serious issues in Light Trail and Light Bus networks and hence fragmentation does not lead to performance degradation. The pseudocode for light bus protocol with fragmentation is given in Appendix B.

An alternate method to improve performance would be to consider the packets to be having the size of 1/4 MTU and allow burst mode transmission that can be of length $1, 2, 3$ or 4 packets. This is a close equivalent of the model proposed here.

6. Conclusions

It is increasingly becoming evident that the conventional network architectures are either inherently unsuitable or inadequate for efficient handling of highly variable, sub-wavelength IP traffic. The flexible dynamic sub-wavelength provisioning features of Light Trails coupled with fast channel set up and tear down enable service providers to support multigranularity traffic and offer high-value services at a rapid rate and at competitive costs. This paper reviews a recently proposed architecture called Light Trails and proposes a variant of it called the Light Bus. The paper designs and analyzes simple medium access control protocols based on carrier sensing for these networks. An exact model of a three-node Light Bus and Light Trail are put forth. The protocols are simulated for a five-node scenario and observed to perform well until about loads of 0.7 C. The Light Bus protocol outperforms the Light Trail protocol when small sized packets are prevalent while the Light Trail protocol fares better when packet aggregation is done. However, the Light Bus architecture has a better node control structure that prevents undesired suspensions and retransmissions of packets and avoids the guard band overhead which is found in the Light Trail protocol. We believe to see significant performance enhancement in the modified Light Bus protocol that allows fragments to be transmitted and we continue to work on this. We are also investigating distributed mechanisms like p-persistence, with p being different for each node, for enforcing fairness into our protocols.

Acknowledgments

The research reported in this paper is funded in part by the National Science Foundation under grant ANI 9973102 and ANI-0087746. We would like to acknowledge Prof. Srinivasan Ramasubramanian, University of Arizona at Tucson, for his valuable contributions to this paper. We also acknowledge Prof. Mani Mina, Iowa State University, for our technical discussions with him.

References

[1] I.Chlamtac and A.Gumaste, *Light-Trails: A solution to IPcentric communication in the optical domain*, in the Second Intl. Workshop on Quality of Service in Multiservice IP Networks (QoS-IP 2003), M. Ajmone Marsan et al (editors), Springer-Verlag Heidelberg, pp. 634-644.

[2] J. Turner, *Terabit burst switching*, J.High Speed Networks (JHSN), vol. 8, no.1, 1999.

[3] L.Kleinrock, *Queueing Systems. Volume I,II.* New York, John Wiley and Sons, 1975.

[4] M. Yoo and C.Qiao, *Optical Burst Switching (OBS) - a new paradigm for the optical internet*, J.High Speed Networks (JHSN), vol 8, no. 1, 1999.

[5] N.K. Jaiswal, *Priority Queues*, New York, Academic Press, 1968.

[6] R.W. Wolff, *Stochastic modeling and the theory of queues*, Prentice Hall, Englewood Cliffs, NJ,1988.

[7] S.Yao, S.J.Ben Yoo, B. Mukherjee, *All-optical packet switching for metropolitan area networks: opportunities and challenges*, IEEE Comm. Mag., Mar 2001.

[8] H.Takagi, *Queueing Analysis. Volume I.* New York, Elsevier Science Publishers, 1991.

Appendix A

Here, we present the derivations of expressions for $\phi_B(z), \phi_{R_E}(z), q$, $\phi_{b_2+S_0}(z)$, $\phi_{S_i}(z)$, and $\phi_{b_2}(\lambda_1 + z)$. We first explain how $\phi_{R_E}(z)$ can be obtained from $\phi_E(z)$. For the extended busy period, we have

$$E = D + B + \sum_{i=0}^{n-1}(B_i + S_i) + B_n \tag{12}$$

with probability $p^n(1-p)$ where the probability of at least one arrival in D time units is

$$p = 1 - e^{-\lambda_1 D} \tag{13}$$

B_i and S_i are sequences of i.i.d random variables with MGFs $\phi_B(z)$ and $\phi_B z$ respectively. So, the MGF of E is

$$\phi_E(z) = \phi_D(z)\phi_B(z)\sum_{n=0}^{\infty}\phi_B(z)^n\phi_S^n(z)p^n(1-p) \tag{14}$$

$$= \phi_D(z)\phi_B(z)\frac{1-p}{1-p\phi_B(z)\phi_S(z)}$$

Since D is a constant, we have

$$\phi_D(z) = e^{-zD} \tag{15}$$

The MGF of the busy period is

$$\phi_B(z) = \phi_{b1}[z + \lambda_1 - \lambda_1 \phi_B(z)] \tag{16}$$

For the silence period, $S_i, i > 1$, we have,

$$\phi_S(z) = \int_0^D \frac{\lambda_1 e^{-\lambda_1 t} e^{-zt}}{1 - e^{-\lambda_1 D}} dt$$

$$= \frac{\lambda_1}{\lambda_1 + z} \frac{1 - e^{-(\lambda_1 + z)D}}{1 - e^{-\lambda_1 D}} \tag{17}$$

$$\phi_E(z) = \frac{e^{-(\lambda_1 + z)D} \phi_{b_1}[z + \lambda_1 - \lambda_1 \phi_B(z)]}{1 - \frac{\lambda_1}{\lambda_1 + z} \phi_{b_1}[z + \lambda_1 - \lambda_1 \phi_B(z)](1 - e^{-(\lambda_1 + z)D})} \tag{18}$$

The mean value of E can be obtained from the first derivative of the MGF of E as

$$\bar{E} = -\phi'_E(0) = \frac{e^{\lambda_1 D} - 1}{\lambda_1} + \frac{\bar{b_1} e^{\lambda_1 D}}{1 - \lambda_1 \bar{b_1}} \tag{19}$$

We obtain the residual lifetime of E using $\phi_E(z)$ and $\bar{E}$ as

$$\phi_{R_E}(z) = \frac{1 - \phi_E(z)}{z\bar{E}} \tag{20}$$

Next, we obtain the value of q from $\bar{E}$ and $\bar{B}$ as follows.

$$\bar{B} = -\phi'_B(0) = \frac{\bar{b_1}}{1 - \lambda_1 \bar{b_1}} \tag{21}$$

The number of packets sent during busy period is given by

$$\frac{\bar{B}}{\bar{b_1}} = \frac{1}{1 - \rho_1}$$

where $\rho_1 = \lambda_1 \bar{b_1}$. The number of packets sent during extended busy period is given by

$$\bar{N} = \frac{1}{1 - \rho_1} + \sum_{n=0}^{\infty} \frac{np^n(1-p)}{1 - \rho}$$

$$= \frac{1}{(1-p)(1-\rho_1)}$$

$$= \frac{e^{\lambda_1 D}}{1 - \lambda_1 \bar{b_1}} \tag{22}$$

We introduce the "virtual" service time of the queue 1 packet to be the modified service time of the packet to account for the extended busy period observed by node 2. Since $\bar{E}$ is the time during which $\bar{N}$ packets were sent, the virtual service time of a packet from queue 1 is

$$\bar{V} = \bar{E}/\bar{N} = \bar{b_1} e^{-\lambda_1 D} + \frac{1 - e^{-\lambda_1 D}}{\lambda_1}$$

$\lambda_1 \bar{V}$ is the probability that queue 1 is active. Hence, we have:

$$q = 1 - \lambda_1 \bar{E}/\bar{N_1} = e^{-\lambda_1 D}(1 - \lambda_1 \bar{b_1})$$

This expression for q can be interpreted as follows. For the delay line to be empty at time t=D, the server at queue 1 should have been idle at time t=0, which happens with probability $(1 - \lambda_1 \bar{b_1})$ and no arrival should have happened on queue 1 for the next D time units which happens with probability $e^{-\lambda_1 D}$.

The expression for $\phi_{b_2}(\lambda_1 + z)$ is given by

$$\phi_{b_2}(\lambda_1 + z) = \int_{t=0}^{t=\infty} e^{-(\lambda_1 + z)t} f_{b2}(t) dt \tag{23}$$

Next, we find an expression for the joint density function of $b_2 + S_0$. b_2 and S_0 are dependent random variables. We have
$f_{b_2+S_0}(b_2 + S_0 = t + v|$ an arrival happens at queue 1 in time $b_2 = t)$

$$= \frac{\lambda_1 e^{-\lambda_1(t+v-D)}}{1 - e^{-\lambda_1 t}}$$

since $u + D = t + v$ as seen in Figure 5.

$$\phi_{b_2+s_0}(z) = \int_{t=0}^{t=D} \int_{v=D-t}^{D} \frac{\lambda_1 e^{-\lambda_1(t+v-D)} e^{-(t+v)z}(1 - e^{-\lambda_1 t})}{(1 - e^{-\lambda_1 t})} f_{b_2}(t) dv dt$$

After some manipulations, we get

$$\phi_{b_2+s_0}(z) = \frac{\lambda_1 e^{zD}}{\lambda_1 + z}[1 - \phi_{b_2}(\lambda_1 + z)] \tag{24}$$

Appendix B

Light Trail Protocol

```
if (Packet.Ready == TRUE)

{

    if (LINE == BUSY)
        WAIT_UNTIL_DELAY_LINE_IDLE;
    if (LINE == IDLE)
    {
        do
        {
            SEND_BEACON_SIGNAL;
            GUARD_BAND_WAIT;
            START_TRANSMISSION(Packet);
            if (UPSTREAM_ACTIVITY_DURING_TX == TRUE)
            {
            ABORT_TRANSMISSION(Packet);
            WAIT_UNTIL_DELAY_LINE_IDLE;
            }
        }
        while (FULL_PACKET_TRANSMITTED == FALSE)
    }

}
```

Light Bus protocol

```
if (Packet.Ready == TRUE)

{

    if (DELAY_LINE == IDLE)

        TRANSMIT(Packet);

    if (DELAY_LINE == BUSY)

    {

        WAIT_UNTIL_DELAY_LINE_IDLE;
        TRANSMIT(Packet);
```

```
        }
    }
```

Light Bus with Fragments protocol

```
    if (Packet.Ready == TRUE)
    {
        do
        {
            if (DELAY_LINE == IDLE)
            {
                COMPUTE(Idle Period, Fragment Size);
                TRANSMIT(Packet Fragment);
                COMPUTE(Remaining Fragment);
            }
            if (DELAY_LINE == BUSY)
                WAIT_UNTIL_DELAY_LINE_IDLE;
        }
        while (FULL_PACKET_TRANSMITTED == FALSE)
    }
```