Design and Analysis of an Asynchronous WDM Local Area Network Using a Master/Slave Scheduler

Eytan Modiano and Richard Barry
MIT Lincoln Laboratory
244 Wood St.
Lexington, MA 02420

Abstract - We describe an architecture and Medium Access Control (MAC) protocol for WDM networks. Our system is based on a broadcast star architecture and uses an unslotted access protocol and a centralized scheduler to efficiently provide bandwidth-on-demand in WDM networks. To overcome the effects of propagation delays the scheduler measures the delays between the terminals and the hub and takes that delay into account when scheduling transmissions. Simple scheduling algorithms, based on a look-ahead capability, are used to overcome the effects of head-of-line blocking. An important application area for this system is in optical *Access Networks*, where this novel MAC protocol can be used to access wavelengths in a WDM Passive Optical Network (PON).

I. Introduction

In recent years there has been a wave of research toward the development of WDM-based Local Area Networks (LANs) [1-10]. Most of the proposed protocols and architectures are based on a broadcast star network architecture. Some of the protocols are based on random access and consequently result in low throughput due to contention [3,4]. Other protocols attempting to minimize contention, through the use of some form of reservations, require that the system be synchronized and slotted, and many of these protocols require multiple transceivers per node [5-8]. Despite the added complexity of these systems, most still fail to achieve high levels of utilization due to inefficient scheduling scheme that fail to deal with receiver contention or ignore the effects of propagation delays. A comprehensive survey of WDM multi-access protocols and their properties is presented in [1,2].

The purpose of the system described in this paper is to achieve good throughput delay characteristics while maintaining simple user terminals. Previous efforts to simplify user terminals involved protocols [9-10] that use fixed tuned receivers or transmitters. However, those protocols limit the number of users to the number of available wavelengths and are hence not scalable. Also, protocols using only a single fixed tuned device are often limited to the use of a random access protocol, resulting in low channel utilization [3,5].

The architecture and protocol described in this paper eliminate the need for slotting and synchronization, yet it results in high utilization in both s and passive optical access networks. The system consists of a simple broadcast-and-select star network. Each user terminal consists of a single transmitter and receiver, both of which are tunable over all data wavelengths and one control wavelength. In addition, an optional, fixed tuned transceiver can also be used for the purpose of communicating on the control channel. The proposed system can support 10's of wavelengths operating at 10Gbps each.

This system is particularly applicable to optical access networks. Future optical access network architectures will use a Passive Optical Network (PON) to connect between the Central Office and end-users [19]. Each PON will need to support 100's of users; hence, there will be a need for users to share wavelengths. The proposed system is ideally suited for providing bandwidth-on-demand in this environment.

The system is novel in a number of ways. First, it uses an unslotted MAC protocol yet results in high efficiency even in high latency environments. The choice of an unslotted protocol is driven by the desire to eliminate the requirement to maintain slotting in the network. Unfortunately, unslotted MAC protocols such as CSMA result in very low utilization when used in systems with high latency. Alternatively, high latency protocols such as unslotted Aloha are limited in throughput to less than 18% [3,5]. Another novelty of our system is that it uses a centralized master/slave scheduler which is able to schedule transmissions efficiently. To overcome the effects of propagation delays the scheduler measures the delays between the terminals and the hub and takes that delay into account when scheduling transmissions.

The system uses simple scheduling algorithms that can be implemented in real-time. Unicast traffic is scheduled using first-come-first-serve input queues and a window selection policy to eliminate head-of-line blocking, while multicast traffic is scheduled using a random algorithm [12]. Analysis and simulations show that the system can achieve low delays even at high loads.

While the use of a centralized scheduler can significantly improve the performance of the system, it also increases the overall cost of the system. However, the functionality of the scheduler described in this paper is relatively simple and can be easily implemented in a single ASIC. The cost of such a scheduler, which is shared among all of the users in the network, is relatively minimal when compared to the overall cost of the network. For this reason, in recent years a number of

0-7803-5417-6/99/$10.00 ©1999 IEEE.

"centralized" systems are being deployed for use in high speed networks. For example, Switched Gigabit Ethernet uses a centralized switch hub and the recent Hybrid-Fiber-Coax (HFC) standard for data transmission over the cable infrastructure uses a centralized scheduler [20].

This paper is organized as follows. In Section II we describe the basic network architecture. In Section III we describe various aspects of the MAC protocol and the associated scheduling algorithms and in Section IV we develop an approximate analysis of delay as well as a simulation model.

II. System architecture

The network consists of optical terminals (OTs) that are connected via a simple broadcast star located at a hub (which can be at the central office in an access network environment). As shown in Figure 1, each OT is connected to the star using two fibers, one in each direction. Transmissions from all OTs on all wavelengths are combined at the star and broadcast to the OTs on the downlink fibers. Each OT is equipped with a single transmitter and receiver, both of which are tunable to all wavelengths, as shown in figure 2A. All OTs send their requests to the scheduler on a dedicated control wavelength, λ_C. The scheduler, located at the star, schedules the requests and informs the OTs on a separate wavelength, $\lambda_{C'}$, of their turn to transmit. Upon receiving their assignments, OTs immediately tune to their assigned wavelengths and transmit. Hence OTs do not need to maintain any synchronization or timing information. By measuring the amount of time that OTs take to respond to the assignments, the scheduler is able to obtain an estimate of each OTs round-trip delay to the hub. This delay information is then used by the scheduler to overcome the effects of propagation delays. Access to the control channel is obtained using a simple version of the unslotted Aloha protocol, as described in section III.

With only a single transceiver per OT, receivers cannot monitor both the control channel and the data channels simultaneously. Therefore, the scheduler cannot send scheduling information to a node that is receiving on one of the data channels. Similarly, a node cannot send reservation requests while it is transmitting on one of the data channels. There are a number of approaches that can be used to overcome this problem. The simplest would be to for the scheduler to only schedule nodes to transmit (or receive) in a time division multiplexing (TDM) fashion, in alternative time slots[1], so that nodes can regularly visit the control channel to send requests and receive scheduling assignments. Alternatively, the scheduler can make sure to schedule transmissions on the control and data channels so that conflicts are avoided. This alternative, however, would require the scheduler to implement a rather sophisticated scheduling scheme.

Clearly, the use of only a single transceiver per node simplifies the user terminal at the expense of a more sophisticated scheduling scheme and a reduction in the transmission capacity available to users (with a TDM scheme, users would be limited to using half of the available slots[2]). In order to allow users full utilization, a second fixed tuned transceiver can be used for the control channel, as shown in figure 2B. With a second transceiver for the control channel, the data and control channels would be independent and nodes will be able to fully utilize the data channels. For the analysis that follows throughout this paper, we will assume that each node is equipped with a separate transceiver for the control channel. This assumption simplifies the analysis and also allows users to achieve higher throughputs. However, nodes that do not need the full throughput of a channel, can use the protocol with a single transceiver that is shared between the control and data channels.

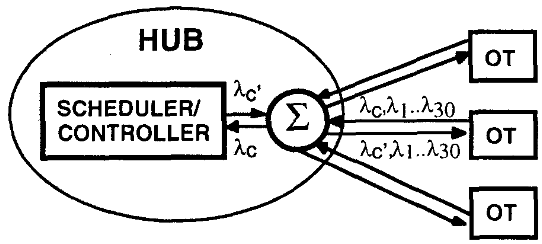

Figure 1. Scheduler based network.

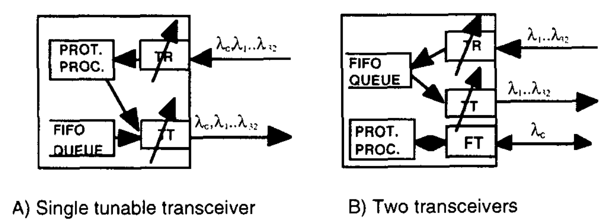

Figure 2. Optical Terminal (OT).

III. Access protocol

Our proposed protocol is based on a simple master/slave scheduler as was shown in figure 1. All OTs send their requests to the scheduler, which schedules the requests and informs the OTs when and on which wavelength to transmit. Upon receiving their assignments, OTs immediately tune to that

[1] Although the scheduler can assign transmissions in a TDM (slotted) fashion, the nodes can remain unslotted and unsynchronized.

[2] Although users will be limited to 50% utilization, the channels may still be fully utilized.

wavelength and transmit. Hence OTs do not need to maintain any synchronization or timing information. There are three major aspects to the protocol. First, the protocol uses ranging to overcome the effects of propagation delays. Second, the protocol uses random access for the control channel. Third, the protocol uses a simple scheduling algorithm with First-Come-First-Serve (FCFS) input queues and a look-ahead window to overcome Head-Of-Line blocking. These are described in more detail below.

A. The use of ranging

The protocol is able to overcome the effects of propagation delays by measuring the round-trip delay of each OT to the hub and using that information to inform OTs of their turn to transmit in a timely manner. For example consider figure 3, in order for OT B's transmission to arrive at the hub at time T, the scheduler must send the assignment to OT B at time T-τ, where τ is OT B's round-trip delay to the hub (including propagation and processing delays). In this way the transmissions of different terminals can be scheduled back-to-back, with little dead-time between transmissions.

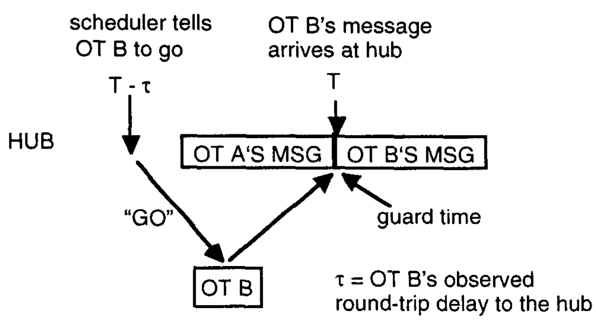

Figure 3. Use of ranging to overcome propagation delays.

An important and novel aspect of this system is the way in which ranging is accomplished. Unlike other systems where terminals need to range themselves to their hubs in order to maintain synchronization [11], here we recognize that it is only the hub that needs to know this range information.. Hence ranging can be accomplished in a straightforward manner. The scheduler ranges each terminal by sending a control message telling the terminal to tune to a particular wavelength and transmit. By measuring the time that it takes the terminal to respond to the request, the scheduler can obtain an estimate of the round trip delay for that terminal. This estimate will also include the tuning and processing delays. Furthermore the scheduler can repeatedly update this estimate to compensate for delay changes (e.g., due to thermal effects[3]). These measurements can also be made by simply monitoring the terminals' response to ordinary scheduling assignments. The significance of this approach is that terminals are not required to implement a ranging function, and this simplifies the end user terminal. A similar approach is employed in [22], where terminals schedule their transmissions to account for propagation delays. In [22], nodes offset their transmissions with the maximum possible propagation delay in order to make sure that both the receiver and transmitter had sufficient time to respond to the control messages.

B. Access to the control channel

Reservations are made using a random access protocol to access the control channel. Terminals send reservation requests repeatedly and update their requests after waiting a random delay. These reservation messages contain the state of the queues at the requesting terminal. For example, each reservation message can contain the destinations with which the terminal wants to communicate and the duration of the requested transmissions.[4]

Reservation requests are sent on the control channel at random, and therefore it is possible for two or more terminals to send their requests during overlapping time intervals. In such a case their transmissions would "collide" and not be received by the scheduler. However, since reservation messages containing the state of the queue are sent repeatedly, all requests will eventually be received by the scheduler. As requests are answered by the scheduler, terminals update their requests to reflect the changes in their request queue.

In order to randomize transmissions on the reservation channel, terminals wait a random, exponentially distributed amount of time, with an average duration $\overline{T}$, between successive transmissions of a reservation request. Notice that unlike a random back-off algorithm, where information about the success or failure of a transmission is available, here we do not rely on any such information but rather repeatedly send the state of the queue. Of course, the state of the queue changes as successful requests are answered by the scheduler[5].

With N terminals and an average rate of one request message every $\overline{T}$ seconds, requests arrive at a rate of $N/\overline{T}$ requests per second. When $\overline{T}$ is much larger than L, the duration of a reservation request, we can model the arrival of requests as

[3] Although the system can compensate for delay variations, in local area and access networks delay variations due to thermal affects will be relatively small and insignificant when compared to the slot duration.

[4] Since sending the complete state information may lead to very large reservation messages, reservation requests may contain only partial information (e.g., first ten requests).

[5] In order to maintain synchronization between the nodes and the scheduler, transmission requests and the scheduling assignments may also have to contain sequence numbers.

Poisson. Using this model, the probability of having n arrivals during a period of time Δ is given by

$$P(n) = \frac{(N\Delta / \overline{T})^n e^{-(N\Delta / \overline{T})}}{n!}.$$

We are interested in computing the average amount of time that is takes a successful request to get through to the scheduler. If it were not for collisions each terminal would get a successful request every $\overline{T}$ seconds. However, due to collisions, some requests will fail and the average amount of time between successful requests will increase. With an unslotted protocol, a request will be successful if no other requests were made in the 2L time period before the end of the transmission. This will happen with probability, $P(0) = e^{-(2L(N-1)/\overline{T})}$, and the average number of transmission attempts per successful transmission is $e^{(2L(N-1)/\overline{T})}$. Therefore, on average, every terminal gets a successful request every Λ seconds, where Λ is given by

$$\Lambda = \overline{T} e^{(2L(N-1)/\overline{T})}.$$

We can now choose $\overline{T}$ to minimize the average access time to the control channel. This can be done by taking the derivative of Λ with respect to $\overline{T}$ and setting it equal to 0.

$$d\Lambda / d\overline{T} = e^{2L(N-1)/\overline{T}} - 2L(N-1)e^{2L(N-1)/\overline{T}} / \overline{T} = 0$$

$$\Rightarrow \overline{T} = 2L(N-1).$$

Hence, setting $\overline{T}$ equal to 2L(N-1) minimizes the access delay and the resulting access delay is $\Lambda_{\min} = 2(N-1)Le$. For example, in a system with 100 nodes, a transmission rate of 10 Gbps, and a control message size of 100 bits, a terminal would send a reservation request on average every 2µs, and the average access delay for a successful reservation would be about 5.5µs.

C. Scheduling algorithm

In order to simplify the design of the scheduler, we use a slotted system where requests are made for fixed size slots and the scheduler maintains a slotted reservation system. However, it is important to note that the OTs remain unslotted and unsynchronized. All of the timing is controlled by the scheduler using the master/slave protocol described in the previous section. As we described in the previous section, in order for the scheduler to schedule transmissions for some slot T, it must send the scheduling information to the nodes one round-trip delay before the start of the slot. This allows the scheduler to compensate for propagation and tuning delays. Therefore, the scheduler must schedule transmissions for a time slot well enough in advance so that all of the nodes scheduled for that slot can be informed. In doing so the scheduler completely overcomes the impact of propagation delay[6].

While the use of ranging can account for tuning delays, the scheduler described in this section assumes fast tuning transceivers. When tuning delays are small compared to slot times, the scheduler is very efficient. The system proposed here will use very fast transceivers that can tune in just a few nano-seconds. While fast tuning transceivers are still largely experimental, they are becoming commercially available and will be used to build the proposed system. However, when tuning delays are large, more complex scheduling algorithms that account for tuning delays can be employed (e.g., [21]).

Even with fast tuning transceivers, the efficient scheduling of transmissions at very high rates is difficult. In a WDM system with a single transmitter and receiver per node, scheduling is constrained by the number of wavelengths, W, which limits the number of requests served during a slot to W. It is also constrained by the fact that each node has a single transmitter and a single receiver. Therefore, during a given slot, each node can be scheduled for at most one transmission and one reception. This, in fact, is a problem very similar to that of scheduling transmissions in an input queued switch. In the case of an input queued switch it is known that when a First-Come-First-Serve service discipline is employed under uniform traffic, throughput is limited to $2-\sqrt{2} = 0.585$ [13]. This throughput limitation is due to the head-of-line (HOL) blocking effect, where transmissions are prevented because the packet at the head of the queue cannot be scheduled due to a receiver conflict. It is also known that if nodes are allowed to look-ahead into their buffers and transmit a packet other than the one at the head of the queue, the effect of HOL blocking can be significantly reduced [14]. Scheduling algorithms based on bipartite graph matching algorithms have been proposed that achieve full utilization under uniform and non-uniform traffic conditions [15,16]. However, it is also known that these algorithms are computationally intensive, requiring $O(N^{2.5})$ operations to be implemented [17].

The network in this paper is being developed to support an enormous traffic volume. For example, with 30 data wavelengths operating at 10 Gbps each and an average slot size of 10,000 bits, 30 million slots have to be scheduled every second. This requirement makes the implementation of a complicated scheduling algorithm impractical with present technology. We therefore resort to a simpler though sub-optimal, algorithm.

---

[6] Clearly with multiple nodes and different tuning delays it may not always be possible for the scheduler to notify all of the nodes just in time for their transmission. This issue can be addressed in a number of ways that are beyond the scope of this paper. For example, a simple solution is for the scheduler to have multiple transmitters. Other more practical approaches also exist.

Our scheduling algorithm is based on input queues. The algorithm is made efficient through the use of a "look-ahead" window that allows the algorithm to look-ahead into each input queue and schedule requests that are not necessarily at the head of their queue. A look-ahead capability of k allows the algorithm to look as far as the k^{th} request in the queue. The algorithm is implemented on a slot-by-slot basis, forming a schedule for the given slot. The algorithm works by maintaining N request queues, each containing the transmission requests from one of the N nodes in the network. The algorithm visits every node in some order (perhaps random, in order to maintain fairness among the nodes) and, starting with the first request in the queue, it searches for a request that can be scheduled. That is, it searches the node's request queue for a transmission request to a receiver that has not been assigned yet. The algorithm searches the queue until depth k has been reached. If a request has been found, a wavelength is assigned to it. This process is continued until either all of the request queues have been visited or all W wavelengths have been assigned. During the next slot, the algorithm starts anew with the first request in each queue. Figure 5 shows an example of the scheduling algorithm with three nodes. Shown in the figure is the destination of each request. After the first request in queue 1 is selected, the second request in queue 2 is selected leaving no available receivers for node 3 to communicate with. Notice, that the algorithm is clearly not maximal in the sense that there are other possible scheduling assignments that would allow all three nodes to transmit (e.g., 1-to-2, 2-to-3, and 3-to-1). Nonetheless, this algorithm improves considerably over an algorithm that looks only at the request at the head of the queue, and it is only slightly more complicated to implement. In fact, it is clear from the description of the algorithm that the algorithm can be implemented in O(kxN) operations, a significant reduction in the number of operations compared to the graph matching algorithms.

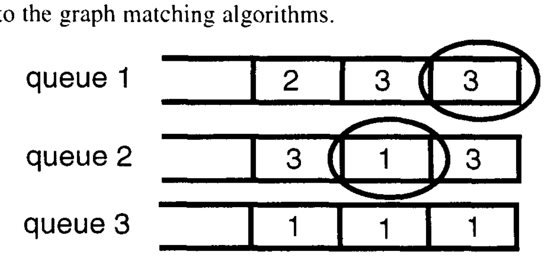

Figure 5. Scheduling algorithm with 3 nodes.

We analyze, through simulation, the maximum throughput that this algorithm can achieve. Table 1 shows the maximum achievable throughput under uniform traffic with 30 data wavelengths. When the number of nodes is equal to the number of channels and no look-ahead is employed (i.e., k=1), HOL blocking limits throughput to 59% as predicted in [13]. However, a look-ahead window of just 4 packets can increase throughput to over 80%. As the number of nodes exceeds the number of channels the effect of HOL blocking is drastically reduced. This is due to two factors; first, the probability that multiple nodes have a packet at the HOL to the same destination is reduced due to the increase in the number of destinations, and second, with fewer channels than nodes the algorithm has many more requests from which to choose a schedule of W transmissions. As can be seen from the table, the combination of more nodes than channels and a look-ahead window of 4 or 5 packets virtually eliminates the effects of HOL blocking on throughput under uniform traffic. Table 2 shows similar results for a system with just 7 data wavelengths.

N	k=1	k=2	k=3	k=4	k=5	k=6	k=7
30	0.59	0.71	0.77	0.81	0.83	0.85	0.86
35	0.69	0.83	0.90	0.94	0.96	0.98	0.99
40	0.79	0.95	0.99	0.99	0.99	0.99	0.99
45	0.89	0.99	0.99	0.99	0.99	0.99	0.99
50	0.96	0.99	0.99	0.99	0.99	0.99	0.99
60	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Table 1. The maximum achievable throughput for a system with 30 wavelengths, N nodes and a look-ahead window k.

N	k=1	k=2	k=3	k=4	k=5	k=6	k=7
7	0.62	0.74	0.79	0.82	0.85	0.86	0.87
10	0.86	0.97	0.99	0.99	0.99	0.99	0.99
14	0.99	0.99	0.99	0.99	0.99	0.99	0.99
21	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Table 2. The maximum achievable throughput for a system with 7 wavelengths, N nodes and a look-ahead window k.

IV. Analysis of Queueing Delay

In order to analyze the average queueing delay in this system we assume that packets arrive to each of the N nodes according to a Poisson random process of rate λ, and are destined, with equal probability, to each of the N nodes[7]. We again assume that all packets are of the same length, take 1 slot to transmit, that the scheduler uses the slotted scheduling described in the previous section, and that all transmissions are scheduled to occur at the beginning of a time slot. As shown in figure 6, the system consists of N nodes and W channels.

[7] Although in a practical system a node would not send a message to itself, this assumption simplifies the notation without a significant impact of the results (especially with large N).

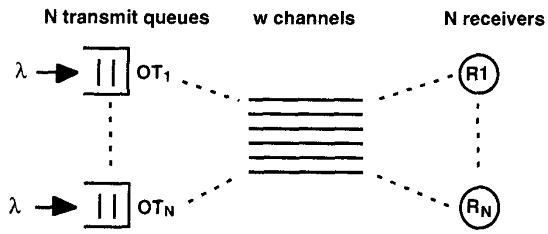

Figure 6. An input queued system with N nodes and W channels.

Clearly in this system the queues at each of the N nodes are dependent on one another, making the analysis of the system difficult. This system can be analyzed using an N^2-dimensional, discrete-time, infinite Markov chain representing the number of requests (packets) between each of the N^2 source/destination pairs[8]. However, obtaining closed form expressions for the steady-state behavior of interacting queues is generally very difficult. Even numerical evaluation can be computationally complex [18]. We are, therefore, forced to consider approximate analysis which we back with simulations.

Our approximate analysis assumes, erroneously, that the N queues are independent of one another. That is, we assume that the probability that the different queues are not empty is independent from queue to queue and that the destinations of the requests at the head of the queues are independent of one another. Both of these assumptions are not correct but allow us to considerably simplify the analysis by focusing on the state of a single queue independently from the rest of the queues.

We start with an analysis of the scheduler without the look-ahead capability (e.g., k=1). Consider the single queue for an arbitrary node i, shown in figure 7, and let X be the amount of time that elapses from the moment that a packet arrives to the head of the queue until it departs from the system. This time includes both the scheduling delay and the transmission delay for the packet at the head of the queue[9]. Clearly, X amounts to the service time of the packet at the head of the queue. Computing the first two moments of X would allow us to apply the well known M/G/1 results for steady-state queueing delays.

[8] Keeping track of queue sizes only is not sufficient because of the receiver contention problem.

[9] Since we assume FCFS scheduling, a packet cannot be scheduled until it arrives at the head of the queue.

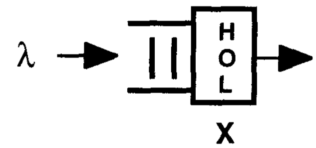

Figure 7. A single node's queue where X represents the amount of time a packet spends in the HOL position.

Let $\overline{X}$ denote the expected value of X. Then the probability that there is a packet at the head of the queue, using Little's law, is given by

$$\rho = \lambda \overline{X}. \tag{1}$$

In order for a packet at the head of the queue to be selected it must find both a free receiver and an available channel (wavelength). Assuming that the queues are examined in random order, the simple FCFS scheduler of the previous section can be described using a two stage process to compute the schedule at the beginning of each slot. In the first stage, the scheduler considers all of the HOL requests from the different queues, and if there are multiple requests for a single receiver it selects one of them at random and "ignores" the rest. In the second stage the scheduler considers all of the requests selected in the first stage and selects at random up to W of them to be transmitted.

Using our independence approximation, we can now proceed to evaluate the probability that a packet at the head of a queue is selected for transmission as follows. We note that the probability a packet is selected is equal to the probability that it is selected in the first stage (by the receiver) times the probability that it is selected in the second stage given that it was selected in the first stage.

Given that a node has a packet to send to a particular destination, we wish to compute the probability that the node's packet is selected in the first stage from among all of the nodes that have a packet to send to the same destination. Each node has a packet to send with probability ρ, and under the uniform traffic assumption, each packet is destined to one of the N-1 receivers with equal probability. Therefore, the probability that a particular node has a packet to send to the same destination is ρ/N. Since there are N-1 other nodes, the probability that i other packets are addressed to the same destination is given by,

$$P_r(i) = \binom{N-1}{i} (\rho / N)^i (1 - \rho / N)^{(N-1-i)}. \tag{2}$$

Now, the probability that the node's packet is selected in the first stage is given by

$$P_{s1} = \sum_{i=0}^{i=N-1} \left(\frac{1}{i+1} \right) \binom{N-1}{i} (\rho / N)^i (1 - \rho / N)^{(N-1-i)}. \tag{3}$$

Thus, after the first stage, the probability that a node has a "selected" packet at the head of its queue is

$$\rho_s = P_{s1} \times \rho.$$

Let N' be the number of nodes selected to proceed to the second stage. Then, using, again, an independence approximation, N' has the following binomial distribution,

$$P(N' = n) = \binom{N}{n} \rho_s^{\,n} (1-\rho_s)^{N-n}. \tag{4}$$

Now, for an arbitrary node x, if it has a packet to send, the probability that it is selected is given by

P(x selected) = P(x selected in first stage and x selected in second stage) = P(x selected in second stage I x selected in first stage) P(x selected in first stage).

We have already computed the probability that node x is selected in the first stage. Given that node x was selected in the first stage, the probability that it will be selected in the second stage is

$$\begin{aligned} P_{s2} = \sum_{i=0}^{w-1} \binom{N-1}{i} \rho_s^{\,i} (1-\rho_s)^{N-1-i} + \\ + \sum_{i=w}^{N-1} \left(\frac{w}{i+1}\right) \binom{N-1}{i} \rho_s^{\,i} (1-\rho_s)^{N-1-i} \end{aligned} \tag{5}$$

Finally, P_s, the probability that node x's packet is selected for transmission is given by

$$P_s = P_{s1} \times P_{s2}, \tag{6}$$

and $\overline{X}$, the average amount of time a packet spends at the HOL, is given by

$$\overline{X} = \sum_{i=1}^{\infty} i(1-P_s)^{i-1} P_s = \frac{1}{P_s}. \tag{7}$$

Now, eq.(7) gives us an expression for $\overline{X}$ in terms of P_s which, in turn, is given by eqs.(3)-(6). However, eqs.(3)-(6) are in terms of ρ, which is given by eq.(1) as a function of $\overline{X}$. For given values of N, W and λ, these equations can be solved iteratively to obtain an approximation for P_s and $\overline{X}$. Finally, we can use the well known M/G/1 formula to obtain the following approximation for average queueing delay:

$$Delay = \overline{X} + \frac{\lambda \overline{X^2}}{2(1-\lambda \overline{X})},$$

where $\overline{X^2}$ is the second moment of X, given by

$$\overline{X^2} = \sum_{i=1}^{\infty} i^2 (1-P_s)^{i-1} P_s = \frac{2-3P_s+P_s^2}{(1-P_s)P_s^2}.$$

The set of iterative equations can be solved to obtain an estimate of the delay using numerical techniques. For simplicity we used the Mathematica programming tool to solve them. The complexity of these iterative equations restricted us to solving only for relatively small values of N. Shown in figure 8 is the predicted delay for N=21 and W=7. Notice that with these values the arrival rate of new packets to a user cannot exceed 1/3 due to the channel constraint. Furthermore, the maximum throughput may be decreased due to the HOL blocking effect, but as can be seen from table 2, the HOL blocking effect on maximum throughput is minimal for these values of N and W. Hence we expect that the maximum achievable arrival rate per node is close to 1/3. As can be seen from the figure the approximation compared extremely well with simulation for low and moderate loads. But, when the load was very high (greater than 0.3), the approximation significantly underestimates delay. This is because at very high loads the interaction between the queues becomes very strong and the independence approximations become inaccurate. This is a common phenomenon in estimating interacting queues, where independence approximations perform well at light to moderate loads and poorly at very high loads [18].

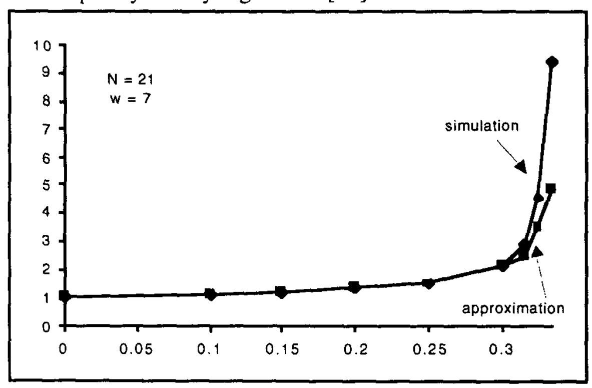

Figure 8. Delay vs. load for a system with 21 nodes and 7 wavelengthsand no look-ahead capability.

The above analysis applies to the scheduler without a look-ahead capability. That is, each of the input queues behaves as a simple FCFS queue. When we introduce the look-ahead capability the analysis becomes significantly more complicated. A simple approximation can be obtained by ignoring receiver contention (i.e., letting $P_{s1} = 1$). This approximation is reasonable when the number of nodes is significantly larger than the number of channels or when the look-ahead is sufficiently large to greatly reduce the effects of receiver contention. For example, looking at tables 1 and 2, all combinations of k and W that result in throughputs of 0.99 seem to be reasonable candidates for applying this simplified approximation. For example, in figure 9 we plot the delays vs. load for a system with 21 nodes, 7 wavelengths and look-ahead values of 1, 2, and 4. Again, as can be seen from the figure, the approximation behaves reasonably well when the load is low to moderate. When the load approaches the maximum of 0.3, the approximation significantly underestimates delay.

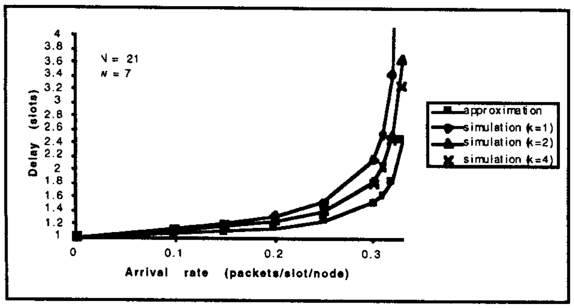

Figure 9. Delay vs. load for a system with 21 nodes and 7 wavelengths and a look-ahead capability (k).

V. Conclusions

This paper describes an architecture and MAC protocol for providing bandwidth on demand in a WDM system. A driving principle in the design is to minimize the cost of the user terminal. To that end, our system uses a single transceiver per node and does not require terminals to be slotted or synchronized. Transmissions are efficiently scheduled using a simple master/slave scheduler located at a hub node. The scheduler is also able to overcome the effects of propagation delays by taking propagation delays into account in the scheduling of transmissions.

This novel system is applicable to high performance local area networks where multi-gigabit per second transmission can be achieved. Another important application area for this system is in optical access networks, where a WDM Passive Optical Network (PON) can be used to provide connectivity between the customer premise and a central office. This MAC protocol, with a scheduler located at the central office, can be used to allow users to share wavelengths over the PON.

References

[1] B. Mukherjee, "WDM-Based Local Lightwave Networks Part I Single-Hop Systems," IEEE Network, May, 1992.

[2] G. N. M. Sudhakar, M. Kavehrad, N.D. Georganas, "Access Protocols for Passive Optical Star Networks," Computer Networks and ISDN Systems, pp. 913-930, 1994.

[3] N. Mehravari, "Performance and Protocol Improvements for Very High Speed Optical Fiber Local Area Networks Using a Passive Star Topology," JLT, April, 1990.

[4] M.S. Chen, N.R. Dono, R. Ramaswami, "A New Media Access Protocol for Packet Switched Wavelength Division Multiaccess Metropolitan Network," JSAC, August, 1990.

[5] H. B. Jeon and C. K. Un, "Contention Based Reservation Protocols in Multiwavelength Protocols with Passive Star Topology," ICC'92.

[6] I. Chlamtac and A. Ganz, "Channel Allocation Protocols in Frequency-time Controlled High-Speed Networks," IEEE Transactions on Communications, April, 1988.

[7] F. Jia and B. Mukherejee, "The Receiver Collision Avoidance (RCA) Protocol for Single hop WDM Lightwave Networks," ICC'92, Chicago, June, 1992.

[8] I. M. I. Habib, M. Kavehrad, C.-E. W. Sundberg, "Protocols for Very High-Speed Optical Fiber Local Area Networks Using a Passive Star Topology," J. Lightwave Technology, Dec., 1987.

[9] G.N.M. Sudhakar, N.D. Georganas and M. Kavehrad, "A Multi-channel Optical Star LAN and its Application as a Broadband Switch," ICC '92, June, 1992.

[10] P.Dowd, "Random Access Protocols for High Speed Interprocess Communications Based on a Passive Optical Star Topology," Journal of Lightwave Technology, June, 1991.

[11] I.P. Kaminow, et. al., "A Wideband All-Optical WDM Network", IEEE JSAC, June, 1996.

[12] E. Modiano, "Unscheduled Multicasts in WDM Broadcast-and-Select Networks," Infocom '98, SF, CA, March, 1998.

[13] M.J. Karol, M.G. Hluchyj and S.P. Morgan, "Input Versus Output Queueing in a Space-Division Packet Switch," IEEE Transactions on Communications, December, 1987.

[14] M.G. Hluchyj and M.J. Karol, "Queueing in High-Performance Packet Switching," JSAC, December 1988.

[15] K. Sivalingam and J. Wang, "Media Access Protocols for WDM Networks with On-Line Scheduling," JLT, June, 1996.

[16] N. McKeown, V. Anantharam and J. Walrand, "Acheiving 100% Throughput in an Input Queued Switch," Infocom '96, San Francisco, CA, April 1996.

[17] J.E. Hopcroft and R.M. Karp, "An $n^{5/2}$ Algorithm for Finding Maximal Matching in Bipartite Graphs," Society for Industrial and Applied Math. Journal of Computing, Feb. 1973.

[18] E. Modiano and A. Ephremides, "A Method for Delay Analysis of Interacting Queues in Multiple Access Systems," Infocom 93, San Francisco, CA, March, 1993.

[19] E. Modiano and R. Barry, "Architectural Considerations in the design of WDM-based Optical Access Networks," to appear *in Computer Networks and ISDN Systems*, Special issue on broadband access. Approximate publication date: Dec., 1998.

[20] S. Perkins and A. Gatherer, "Two-way Broadband CATV-HFC Networks: State of the art and Future trends," to appear *in Computer Networks and ISDN Systems*, Special issue on broadband access. Approximate publication date: December, 1998.

[21] F. Jia, B. Mukherjee and J. Iness, "Scheduling Variable-Length Messages in a Single-hop Multichannel Local Lightwave Network," IEEE/ACM Transactions on Networking, August, 1995.

[22] F. Jia and B. Mukherjee, "Performance Analysis of a Generalized Receiver Collision Avoidance (RCA) Protocol for Single-hop WDM Local Lightwave Networks," Proc. Of SPIE, Vol. 1784, pp. 229-40, 1992.