HiPeR-ℓ: A High Performance Reservation Protocol with ℓook-ahead for Broadcast WDM Networks *

Vijay Sivaraman *George N. Rouskas*
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206

Abstract

We consider the problem of coordinating access to the various channels of a single-hop WDM network. We present HiPeR-ℓ, a new reservation protocol specifically designed to overcome the potential inefficiencies of operating in environments with non-negligible processing, tuning, and propagation delays. HiPeR-ℓ differs from previous reservation protocols in that each control packet makes reservations for all data packets waiting in a node's queues, thus significantly reducing control overhead. Packets are scheduled for transmission using algorithms that can effectively mask the tuning times. HiPeR-ℓ also uses pipelining to mask processing times and propagation delays. We use Markov chain theory to obtain a necessary and sufficient condition for the stability of the protocol. The stability condition provides insight into the factors affecting the operation of the protocol, such as the degree of load balancing across the various channels, and the quality of the scheduling algorithms. The analysis is fairly general, as it holds for MMBP-like arrival processes with any number of states, and for non-uniform destinations.

1 Introduction

Wavelength Division Multiplexing (WDM) is the most promising technology for bridging the gap between the speed of electronics and the virtually unlimited bandwidth available within the optical medium. One of the candidate WDM architectures for implementing a new generation of high speed communication networks is the single-hop architecture [9]. In a single-hop network, both a transmitter at the source and a receiver at the destination must operate on the same wavelength for a successful packet transmission. Thus, the problem of coordinating access to the various wavelengths of the network arises. This problem is further complicated by the fact that, at high data rates, propagation delays, processing times, and transceiver tuning times all become non-negligible, and may actually be significantly larger than the packet transmission time. A number of reservation protocols for single-hop networks have appeared in the literature; we review some of these protocols in the next section.

In this paper, we present HiPeR-ℓ, a new reservation protocol for single-hop WDM local area networks. The novelty of HiPeR-ℓ lies in the fact that, by transmitting a single control packet, nodes can make reservations for multiple data packets. Thus, control overhead is significantly reduced, and nodes can use scheduling algorithms that can effectively mask tuning times [12]. HiPeR-ℓ also uses pipelining to mask processing times and propagation delays; parameter ℓ (the *look-ahead*) of the protocol controls the degree of pipelining. Drawing upon results from Markov chain theory, we obtain a necessary and sufficient condition for the stability of the protocol that provides insight in the factors affecting the protocol's operation. In the analysis, we assume arrival processes that capture the notion of burstiness and the correlation of interarrival times, two important characteristics of traffic in high speed networks [11].

In the next section, we review some of the media access protocols for single-hop WDM networks. In Section 3 we present the network and traffic model, and in Section 4 we describe HiPeR-ℓ. In Section 5 we carry out a stability analysis of HiPeR-ℓ. In Section 6 we present some numerical results, and we conclude the paper in Section 7.

2 Why A New Multiple Access Protocol?

Access to the various channels of a single-hop network is usually based on reservation schemes that require the use of control channels [4, 5, 7, 6]. Existing protocols require that control information be transmitted on the control channel for *each* packet sent on the data channels. Typically, TDMA is employed in the control channel with a control slot consisting of N mini-slots, one for each of the N nodes in the network.

In *tell-and-go* protocols [4, 6] the data packet is sent on the node's home channel immediately after the transmission of the corresponding control information. Thus, receiver collisions may arise and explicit acknowledgments are needed. Other protocols are *tell-and-wait* in nature [5, 7, 15]; in other words, nodes send the control information and wait for the control slot to reach all receivers. Then, they process the information in the control slot to determine if a data slot has been reserved for them. In the event of a successful reservation, the packet is transmitted in the corresponding slot and channel. In effect, the control slot information in tell-and-wait schemes is used by the individual nodes to build a picture of the packet queues at all other nodes in the network.

The above protocols suffer from two problems. First, the control channel represents an *electronic processing bottleneck* [6] as control information for N packets must be received and processed for *each packet transmission and reception*. Secondly, all these control channel protocols operate by schedul-

*This work was supported in part by a grant from the Center for Advanced Computing and Communication, NC State University.

0-8186-7780-5/97 $10.00 © 1997 IEEE

ing a *single* packet from each transmitter at a time (typically, the head-of-line packet is the one scheduled for transmission). This packet is scheduled *independently* of other packets waiting for transmission at the same node. Hence, depending on the protocol, one transmitter or receiver tuning time is incurred for each packet transmission/reception.

A protocol that overcomes the processing bottleneck by introducing $k > 1$ control channels was presented in [6]. The main drawback of this protocol, however, is its lack of scalability, as it requires $N + k$ wavelengths. The PROTON protocol [7] can operate with any number of wavelengths, and its design explicitly considers tuning and processing times. However, PROTON schedules one packet at a time, and the results in [7] confirm the intuition that high processing and tuning times have a significant effect on delay and throughput. The MaTPi protocol [16] also considers tuning times, and uses pipelining to mask their effect.

Distributed Queue Multiple Wavelength (DQMW) [8] is another protocol that can operate with any number of wavelengths and which considers tuning times when scheduling packets. DQMW attempts to overcome the head-of-line blocking of other media access schemes by considering multiple packets for transmission by a given node. But these packets are scheduled independently of each other, thus a tuning overhead is incurred for each. In addition, this protocol has higher processing requirements compared to other protocols, as two control packets must be sent for each data packet: one before its transmission and one after the end of its transmission. FatMAC [14] is a reservation protocol that does not require a separate control channel. Instead, all channels operate in cycles, with each cycle consisting of a control and data phase. Reservations are transmitted in the control phase, and the corresponding data packets are sent in the following data phase. As in other protocols, reservations are made only for the head-of-line packets, thus a control and tuning overhead is incurred for each data packet.

In this paper we present HiPeR-ℓ, a new reservation protocol that has the following important features: (a) t is scalable, as it can operate with any number of channels $C \leq N$; (b) it may operate without a control channel, thus all channels are available for data transmission and no extra hardware is needed to monitor and access a control channel; (c) it requires tunability only at one end; (d) it ensures that packet transmissions are free of channel and receiver collisions. (e) it schedules multiple packets for transmission by a node on a given channel to mask the tuning latency [12] and to keep the control requirements low; (f) it uses pipelining to overlap processing with packet transmissions, and to hide the effects of propagation delay.

3 System and Traffic Model

3.1 Network Model

We consider an optical broadcast WDM network with N nodes, each employing one transmitter and one receiver. There are C wavelengths in the network, $\lambda_1, \cdots, \lambda_C$, with $C \leq N$. There is no separate control channel; all channels are used for data transmission, as well as for communicating control information. Without loss of generality, we only consider tunable-transmitter, fixed-receiver (TT-FR) networks. Each tunable transmitter can tune to, and transmit on any wavelength. The fixed receiver at station j, on the other hand, is assigned a home channel $\lambda(j) \in \{\lambda_1, \cdots, \lambda_C\}$. We let $\mathcal{R}_c$ denote the set of receivers sharing wavelength λ_c: $\mathcal{R}_c = \{j \mid \lambda(j) = \lambda_c\}$.

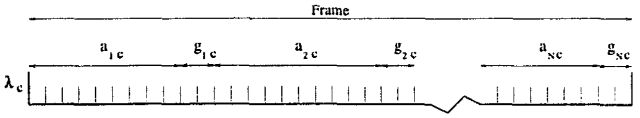

Figure 1: Part of the schedule corresponding to channel λ_c

The network is packet-switched, with fixed-size packets. The buffer space at each node is partitioned into C independent queues. Each queue contains packets destined for receivers which listen to a particular wavelength. This arrangement eliminates the head-of-line blocking problem, and permits a node to send a number of packets back-to-back when tuned to a particular channel. The network operates in a slotted mode, with a slot time equal to a packet transmission time. All nodes are synchronized at slot boundaries. Packets buffered at the c-th queue of each node are transmitted on a FIFO basis into the optical medium on wavelength λ_c.

We let integer $\Delta \geq 1$ denote the number of slots a tunable transmitter takes to tune from one wavelength to another. We also let τ denote the one way propagation delay between a pair of nodes. Without loss of generality, we take τ to be the same for all source-destination pairs in the network.

3.2 Transmission Schedules

One of the potentially difficult issues that arise in a WDM environment is that of packet scheduling in the presence of non-negligible tuning latencies. The authors have recently considered the problem of packet scheduling with tuning latencies in [12]. That work expanded upon, and generalized earlier results obtained in [1, 3]. More specifically, we have shown that careful scheduling can mask the effects of arbitrarily long tuning latencies. The key idea is to have each tunable transmitter send a *block* of packets on each wavelength before switching to the next. Doing so makes it possible to overlap the tuning latency at a node with packet transmissions from other nodes. The main result of [12] was a set of new algorithms for constructing near-optimal (and, under certain conditions, optimal) schedules for transmitting a set of traffic demands $\{a_{ic}\}$. Quantity a_{ic} represents the number of packets to be transmitted by node i onto channel λ_c. The schedules are such that no collisions ever occur. They are also easy to implement in a high speed environment, since the order in which the various nodes transmit is the same for all channels [12].

Figure 1 illustrates the part of such a schedule corresponding to channel λ_c. Each node i is assigned a_{ic} *contiguous* slots for transmitting packets on that channel. These a_{ic} slots are followed by a *gap* of $g_{ic} \geq 0$ slots during which no node may transmit on λ_c. This gap may be necessary to ensure that node $i+1$ has sufficient time to tune from wavelength λ_{c-1} before starting transmission on λ_c. However, the algorithms in [12] are such that the number of slots in most of the gaps is equal to either zero or a small integer. Thus, the length of the schedule is very close to the lower bound.

The scheduling algorithms we have developed require complete information about the traffic demands $\{a_{ic}\}$. HiPeR-ℓ is a reservation protocol that allows the network nodes to dynamically share this information.

3.3 Traffic Model

The arrival process to each node is characterized by a two-state Markov Modulated Bernoulli Process (MMBP), hereafter referred to as 2-MMBP. This is a Bernoulli process whose arrival rate varies according to a two-state Markov chain. It captures the notion of burstiness and the correlation of successive interarrival times, two important characteristics of traffic in high-speed networks. For details on the properties of the 2-MMBP, the reader is referred to [10]. We assume that the arrival process to node $i, i = 1, \cdots, N$, is given by a 2-MMBP characterized by the transition matrix $\mathbf{Q}_i$, and by $\mathbf{A}_i$ as follows:

$$\mathbf{Q}_i = \begin{bmatrix} q_i^{(00)} & q_i^{(01)} \\ q_i^{(10)} & q_i^{(11)} \end{bmatrix} \; ; \; \mathbf{A}_i = \begin{bmatrix} \alpha_i^{(0)} & 0 \\ 0 & \alpha_i^{(1)} \end{bmatrix} \quad (1)$$

In (1), $q_i^{(kl)}, k, l = 0, 1$, is the probability that the 2-MMBP will make a transition to state l, given that it is currently at state k. Obviously, $q_i^{(k0)} + q_i^{(k1)} = 1, k = 0, 1$. Also, $\alpha_i^{(0)}$ and $\alpha_i^{(1)}$ are the arrival rates of the Bernoulli process at states 0 and 1, respectively. We assume that the arrival process to each node i is given by a different 2-MMBP, independent of the arrival processes to other nodes. From [10] we obtain the average arrival rate γ_i of the i-th 2-MMBP as:

$$\gamma_i = \frac{q_i^{(10)} a_i^{(0)} + q_i^{(01)} a_i^{(1)}}{q_i^{(01)} + q_i^{(10)}} \quad (2)$$

We note that γ_i is the probability that any slot contains a packet, regardless of the state of the 2-MMBP. We will only consider 2-MMBPs for which:

$$q_i^{(kl)} > 0, \quad k, l = 0, 1, \quad i = 1, \cdots, N \quad (3)$$

Conditions (3) guarantee that the two-state Markov chain of each 2-MMBP is irreducible and aperiodic, thus it has a stationary distribution.

We let r_{ij} denote the probability that a new packet arriving to node i will have j as its destination node. We will refer to $\{r_{ij}\}$ as the *routing probabilities*. This description implies that the routing probabilities are source node dependent and non-uniformly distributed.

4 Description of the HiPeR-ℓ Protocol

We now present HiPeR-ℓ, a new reservation protocol that nodes in a single-hop WDM network can use to coordinate access to the various channels. The operation of HiPeR-ℓ is rather simple:

- Each network node periodically sends control packets informing all other nodes about its traffic demands.
- Each node has a copy of the packet scheduling algorithm developed in [12]. Upon receipt of all control packets transmitted by other nodes, each node independently runs the algorithm to determine at what time slots to transmit its own data packets. Since all nodes use the same algorithm and the same input values (obtained from the control packets), no channel or receiver collisions arise.

There are two main differences between HiPeR-ℓ and any of the protocols that have appeared in the literature. First, in HiPeR-ℓ a node does not send a reservation request for its head-of-line packet only. Instead, each control packet of a node i contains information about *all* the packets that were queued in any of i's C queues at a certain instant in time. By sending a control packet, node i is in effect making reservations for all packets it had waiting for transmission at that instant. The next time node i is scheduled to transmit on wavelength λ_c, it will send a number of data packets back-to-back equal to the number of reservations it made for this channel in the corresponding previous control packet. Secondly, control packets are not transmitted over a separate channel. Reservations are *in-band* over the same channels used for data. Furthermore, time in the channels is not divided into distinct reservation and data phases as in FatMAC [14]. Exactly when control packets are transmitted will be discussed shortly.

The next subsection describes a first version of HiPeR-ℓ. We then extend the protocol by introducing pipelining to mask the effects of long propagation delays and processing times.

4.1 The Basic Idea: HiPeR-1

The basic operation of HiPeR-ℓ is illustrated in Figure 2. For reasons that will become apparent shortly, we will refer to this version of the protocol as HiPeR-1.

Assume that, somehow, each node i has made reservations for $a_{ic}^{(k)}$ data packets on wavelength λ_c, and that these reservations are known to all nodes. Each node independently runs the scheduling algorithm in [12] to compute a packet transmission schedule. However, the input to this algorithm is not quantities $\{a_{ic}^{(k)}\}$, but rather quantities $\{a_{ic}^{(k)} + 1\}$; the extra slot is for transmitting a control packet. The algorithm will allocate $a_{ic}^{(k)} + 1$ *contiguous* slots to node i for transmission to destinations listening on wavelength λ_c. We will call this allocation of slots to source-wavelength pairs a *frame*.

Suppose now that at time t_k in Figure 2 all nodes have constructed the k-th frame from the known quantities $\{a_{ic}^{(k)} + 1\}$.

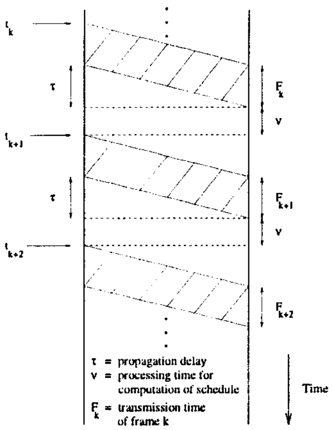

Figure 2: Operation of HiPeR-ℓ when the look-ahead $\ell = 1$

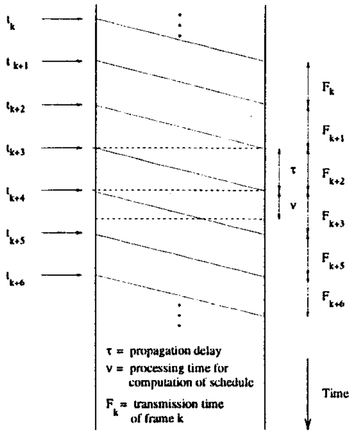

Figure 3: Operation of HiPeR-ℓ when the look-ahead $\ell = 4$

Transmission of this frame can then begin at time t_k. Consider the $a_{ic}^{(k)} + 1$ slots in the frame allocated to node i for transmissions on channel λ_c. Node i will transmit only $a_{ic}^{(k)}$ data packets in these slots (this is the number of data slots it had reserved). In the last slot node i will transmit a control packet with information about the number of data packets that were in its C queues at the beginning of the frame (i.e., at time t_k), excluding packets it transmits during this frame. In other words, a control packet from node i in frame k carries C integers, $a_{i1}^{(k+1)}, \cdots, a_{iC}^{(k+1)}$, and is used to make reservations for future transmissions on each channel. An identical copy of the control packet is transmitted by node i on each wavelength, and carries a special address recognized by all receivers in the network. As a result, by the time the last packet of the frame reaches all receivers, each node has complete information (although somewhat dated) of the queue status at all nodes. Each node can then use this information to run the scheduling algorithm anew to determine the *next* frame, as discussed above.

Let F_k be the length, in slots, of the k-th frame; F_k includes the Δ slots required for tuning the transmitters to their initial channels. Referring to Figure 2 we note that at time $t_k + F_k + \tau$ all nodes will have access to the control information transmitted in frame k (recall that τ denotes the propagation delay). Let ν denote the time it takes to run the scheduling algorithm to construct the next frame [1]. At time $t_{k+1} = t_k + F_k + \tau + \nu$, the transmission of frame $k + 1$ may start. At the same time, each node i will record the number of packets in each of its C queues, and will use that information for constructing its control packets for frame $k + 1$. In effect, the value of a_{ic} in a control packet transmitted in frame $k + 1$ represents the number of packets that arrived to the c-th queue of node i between time t_k (the beginning of transmission of frame k) and time t_{k+1} (the beginning of transmission of frame $k + 1$).

As described, the protocol is said to have a *look-ahead* $\ell = 1$, since control information transmitted during the k-th frame is used to construct the $(k + 1)$-th frame; thus the name HiPeR-1. This protocol falls into the class of *gated* reservation schemes [2], since only those packets that arrived prior to the beginning of frame k will be transmitted in frame $k+1$. HiPeR-ℓ does not have a distinct reservation phase. Instead, control packets are transmitted within a frame along with data packets. This is necessary in order to minimize the tuning overhead. If there was a separate reservation phase, the transmitters would have to (a) tune to each channel during the reservation phase to transmit a single control message, and (b) tune to each channel during the data phase to transmit the data packets.

4.2 Masking Processing and Propagation Delays Through Pipelining

Observe in Figure 2 that there are no transmissions in an interval of size $\tau+\nu$ between the end of frame k (at time t_k+F_k) and the beginning of frame $k + 1$ (at time t_{k+1}). If quantity $\tau + \nu$ is small compared to the average transmission time of a frame, a system running HiPeR-1 will achieve a reasonable throughput. In a high data rate environment, however, processing and propagation delays may be significantly long. As a result, the basic protocol of Figure 2 will experience long idle times with severe effects on overall throughput. We now show how pipelining can solve this problem and keep channel utilization at high levels.

Pipelining can be introduced in the protocol by using values of look-ahead greater than one. Figure 3 illustrates the operation of HiPeR-ℓ when the look-ahead $\ell = 4$. Let us consider frame $k + 1$ whose transmission starts at time t_{k+1}. Control packets transmitted within this frame carry information about the number a_{ic} of data packets that arrived to the various queues in the interval $[t_k, t_{k+1})$. However, this information is not used for constructing frame $k + 2$. As we see in Figure 3, the information carried by the control packets

[1] One important aspect of the scheduling algorithms in [12] is that their running time depends only on system parameters such as the number of nodes and channels, *not* on the actual frame length.

transmitted in frame $k+1$ has not been processed until after time t_{k+4} when frame $k+4$ starts. Thus, this information is used to construct frame $k+5$ whose transmission starts at time t_{k+5}. In general, we have the following rule:

> When the look-ahead is $\ell \geq 1$, the control packets of each frame k carry information about the data packets that arrived during the previous frame $k-1$. This information is used to construct frame $k+\ell$.

As Figure 3 indicates, by selecting an appropriate value for the look-ahead ℓ, we can ensure that a frame is ready for transmission immediately after the end of the previous frame, thus keeping channel utilization at high levels. Let $\bar{F}$ denote the *average* frame transmission time. Then, the value of the look-ahead should be selected as

$$\ell = \left\lceil \frac{\tau + \nu}{\bar{F}} \right\rceil \qquad (4)$$

HiPeR-ℓ incurs an overhead of NC control packets for each frame transmitted (each node sends one control packet on each wavelength). In terms of efficiency, this overhead is not expected to be a problem except at very low data rates when a frame may carry a small number of data packets. On the other hand, the advantage of *in-band* reservation messages over control channel-based reservation schemes is that all available wavelengths can be used to transmit data, and no extra hardware is needed to monitor and access the control channel. However, HiPeR-ℓ can be easily adapted to use *out-of-band* reservation messages (if this is necessary). In this case, for each frame of data packets a node needs to send exactly one control packet on the control channel (as opposed to one control packet for each data packet required by existing protocols). Thus, only a small fraction of the control channel capacity is needed for reservation messages; the remaining capacity can be used for other purposes, such as network management, synchronization, etc.

5 Performance Analysis

An analysis of TDMA schemes in which a node is allocated multiple consecutive slots per frame has been carried out in [13]. There, the generating functions of the queue size and of the delay distribution are derived for fairly general arrival processes. The model in [13] assumes a fixed TDMA frame size, with each node receiving a fixed number of slots occupying the same positions in every frame. Because of the stochastic nature of our system, however, each node will make reservations for, and it will be allocated, a different number of slots from frame to frame. Consequently, the frame size will vary. Furthermore, the scheduling algorithm is run anew for each frame, therefore, the order in which the various nodes transmit may be different in consecutive frames. As a result, the techniques developed in [13] are not applicable here.

For the same reasons, an exact delay analysis of a system running HiPeR-ℓ appears to be difficult. We note, however, that packet delay is directly related to the frame size. In the following, we carry out a stability analysis of HiPeR-ℓ and obtain a necessary and sufficient condition on the total arrival rate to the network for the frame size to remain bounded. Although in our analysis we assume that the arrival process to each node is described by a 2-MMBP, it can be easily seen that the same condition applies to other MMBP-like processes with a larger number of states.

Before we proceed, we note that there are two factors that directly affect the operation of a network running HiPeR-ℓ: the degree of load balancing across the various channels, and the quality of the scheduling algorithm used. In order to quantify their effect on the performance of the protocol, we define two parameters, as follows:

- *Degree of load balancing* $\epsilon_b \geq 0$. Let A_k be the total number of data packets (traffic load) arriving to the network nodes within frame k. Each of these packets will be transmitted on one of the C channels in a future frame. If the load is perfectly balanced across the C channels, each channel will carry exactly $\frac{A_k}{C}$ of these packets. In general, the traffic load will not be perfectly balanced. Parameter ϵ_b is defined so as to provide an upper bound on the number of packets to be carried by any single channel. Specifically, for any frame k, no more than $(1+\epsilon_b)\frac{A_k}{C}$ of the packets arriving during that frame are destined for any given channel. Under perfect load balancing, $\epsilon_b = 0$. The degree of load balancing ϵ_b can be controlled if slowly tunable, rather than fixed receivers are used. Then, as the traffic pattern changes, the network can be reconfigured, i.e., nodes may be assigned new receive wavelengths, so as to keep the load evenly spread across all channels.
- *Scheduling guarantee* $\epsilon_s \geq 0$. Let $\hat{F}_k$ be the lower bound on the length of frame k, based on the data reservations made in a previous frame. Parameter ϵ_s is defined such that, the algorithm used to schedule packet transmissions will always construct a frame of length at most $(1+\epsilon_s)\hat{F}_k$. Under optimal scheduling, $\epsilon_s = 0$.

5.1 Markov Chain Model and Analysis

Consider a network running HiPeR-ℓ with a look-ahead $\ell \geq 1$. We will call a collection of $\ell+1$ consecutive frames a *superframe*. Our analysis below is based on the observation that the data packets transmitted within a superframe are exactly those packets that arrived to the various network nodes during the previous superframe.

We analyze the system by constructing its underlying Markov chain (MC) embedded at superframe boundaries. We observe the system at an instant just before the beginning of a new superframe. The state of the system is described by the tuple $(x, \underline{y})$, where x represents the length, in slots, of the superframe that is about to be transmitted ($x = 0, 1, 2, 3, \cdots$), and $\underline{y}$ is a vector $\underline{y} = (y_1, \cdots, y_N)$, with y_i indicating the state of the arrival process to node i ($y_i = 0, 1,\ i = 1, \cdots, N$).

As the state of the system evolves in time, it defines a MC $\mathcal{M}$. To see this, let $(x, \underline{y})$ be the current state of the system, and $(x', \underline{y}')$ be the state at the beginning of the next superframe. Obviously, the new state $\underline{y}'$ of the arrival processes depends only on the current state $\underline{y}$ and the number of slots x that will elapse. The length x' of the new superframe depends on (a) the number of arrivals during the current superframe and how these packets are distributed across the various channels, (b) the number of control packets to be transmitted within the superframe, and (c) the scheduling algorithm used. The number of arrivals in the current superframe depends only on the state $\underline{y}$ of the arrival processes at the beginning of the superframe, and its length x. The number of control packets transmitted within a superframe is $(\ell+1)CN$, since the superframe consists of $\ell+1$ individual frames. The scheduling algorithm used is independent of the system state. Therefore, the new length x' also depends only on the current state $(x, \underline{y})$.

Let $P[(x, \underline{y}) \rightarrow (x', \underline{y}')]$ denote the probability that the system makes a transition to state $(x', \underline{y}')$, given that it is currently in state $(x, \underline{y})$. (Given the description of the N 2-MMBPs, the value ℓ of the look-ahead, and the scheduling algorithm, the transition probabilities are completely specified. However, the exact values of these probabilities are not necessary in our analysis.) Because of conditions (3), MC $\mathcal{M}$ is irreducible and aperiodic. Thus, $\mathcal{M}$ has a stationary distribution if we can find scalars $\{\pi_{(x,\underline{y})}\}$ that satisfy:

$$\pi_{(x,\underline{y})} = \sum_{(x',\underline{y}')} \pi_{(x',\underline{y}')} P[(x', \underline{y}') \rightarrow (x, \underline{y})], \qquad \forall\, (x, \underline{y}) \quad (5)$$

and such that $\sum_{(x,\underline{y})} \pi_{(x,\underline{y})} = 1$.

Solving the equations (5) by inspection requires writing out the actual values of the transition probabilities, a complicated task. However, we are only interested in obtaining a condition for MC $\mathcal{M}$ to have a stationary distribution. We now observe that random variable $\underline{y}$ can take exactly $K = 2^N$ values which we will denote by $\underline{y_1}, \cdots, \underline{y_K}$. Let us partition the state space of MC $\mathcal{M}$ into subsets S_x of states with the same superframe length: $S_x = \{(x, \underline{y_1}), \cdots, (x, \underline{y_K})\}$. We construct a new MC $\mathcal{M}'$ embedded at superframe boundaries, with state space $\{S_x\}$, and transition probabilities $P_{x,x'}$, where $P_{x,x'}$ is equal to equal to the transition probability in MC $\mathcal{M}$ from the states in S_x to the states in $S_{x'}$. It is now easy to prove the following lemma.

Lemma 5.1 *MC $\mathcal{M}$ has a stationary distribution if and only if MC $\mathcal{M}'$ also has a stationary distribution.*

We are now ready to prove our main result.

Lemma 5.2 *MC $\mathcal{M}'$ has a stationary distribution iff*

$$\gamma < \frac{C}{(1+\epsilon_b)\,(1+\epsilon_s)} \quad (6)$$

where $\gamma = \sum_{i=1}^{N} \gamma_i$ is the total arrival rate to the network.

Proof. Let D_x denote the drift at state x of $\mathcal{M}'$. Because of Pake's lemma [2, 3A.5], in order to show that $\mathcal{M}'$ has a stationary distribution, we only need to show that there exist a state $x_0 \geq 0$ and a scalar $\delta > 0$ such that:

$$D_x \leq -\delta, \qquad \forall\, x > x_0 \quad (7)$$

The drift at state x of MC $\mathcal{M}'$ can be written as:

$$D_x = E[x' \mid x] - x \quad (8)$$

where $E[x' \mid x]$ is the expected length of the next superframe given that the length of the current superframe is x slots.

The expected number of packets that arrive in the current superframe of size x slots, independently of the state of the arrival processes at the beginning and end of the superframe, is γx, where γ is the sum of the arrival rates to the network nodes. Because of the definition of parameter ϵ_b, no more than $(1+\epsilon_b)\frac{\gamma x}{C}$ of these arriving packets are destined for any given channel. In addition, there are $(\ell+1)N$ control packets that will be transmitted on each wavelength within the next superframe. Therefore, the expected number of packets (data plus control) transmitted on any channel during the next superframe cannot be greater than $(1+\epsilon_b)\frac{\gamma x}{C}+(\ell+1)N$. Because of the definition of parameter ϵ_s, the length of this next superframe cannot be greater than $(1+\epsilon_s)$ times this last quantity. Therefore, we can bound the expected length of the next superframe by:

$$E[x' \mid x] \leq (1+\epsilon_s)\,\frac{(\ell+1)NC + (1+\epsilon_b)\,\gamma\, x}{C} \quad (9)$$

If we substitute this expression in (8), we obtain an upper bound on the value of the drift at state x:

$$D_x \leq (1+\epsilon_s)\,\frac{(\ell+1)NC + (1+\epsilon_b)\,\gamma\, x}{C} - x \quad (10)$$

After some algebraic manipulation of (10), we find that (7) is satisfied if we let

$$x_0 = \left\lceil \frac{\delta + (1+\epsilon_s)\,(\ell+1)N}{1 - (1+\epsilon_b)\,(1+\epsilon_s)\,\frac{\gamma}{C}} \right\rceil \quad (11)$$

This x_0 is positive if and only if (6) holds. □

Finally, by combining Lemmata 5.1 and 5.2 we obtain the desired result:

Corollary 5.1 *MC $\mathcal{M}$ has a stationary distribution if and only if the total arrival rate to the network satisfies (6).*

The stability condition (6) is simple yet powerful, as it provides insight into the two main factors that determine the performance of the network, namely, the degree of load balancing, and the quality of the scheduling algorithm. As we can see, the lower the degree of load balancing (i.e., the larger the value of ϵ_b in (6)), the lower the maximum arrival rate that the network can sustain (recall that C is the capacity of the network). Similarly with the scheduling efficiency, captured by parameter ϵ_s in (6). Although (6) was derived specifically for HiPeR-ℓ, we believe that these two factors play a similar role in any reservation protocol for single-hop networks.

Let $\bar{F}$ denote the mean frame size when the stability condition (6) is satisfied. From the definition of the look-ahead ℓ, a packet arriving during a frame k will be transmitted to its destination within frame $k + \ell + 1$. We can then obtain the following expression for the mean packet delay $\bar{D}$:

$$\bar{D} \;=\; (\ell + 1)\,\bar{F} \qquad (12)$$

6 Numerical Results

We demonstrate the operation of the HiPeR-ℓ protocol by considering a client-server network with $N = 40$ nodes and $C = 10$ channels. There are two servers (nodes 1 and 2) and 38 clients (nodes $3, \cdots, 40$). The routing probabilities are:

$$r_{ij} \;=\; \begin{cases} 0 & i = j \\ 0.01 & i = 1, j = 2 \text{ or } i = 2,\ j = 1 \\ \frac{0.99}{38} & i = 1, 2,\ j = 3, \cdots, 40 \\ 0.114 & i = 3, \cdots, 40\ , j = 1, 2 \\ \frac{0.772}{38} & i, j = 3, \cdots, 40 \end{cases} \qquad (13)$$

The arrival process to each of the nodes of the network is described by a different 2-MMBP. The total arrival rate to the network is $\gamma = 7.344$. The arrival processes were selected so that the arrival rate and squared coefficient of variation of the 2-MMBPs take a wide range of values. Based on the results of the previous section, we have assigned receive wavelengths to the various nodes so as to spread the traffic evenly across the various channels. Since there is more traffic entering the two servers, we have decided to assign one wavelength to each of the two servers, while the remaining 8 wavelengths are shared by the other 38 nodes (six of these wavelengths are each shared by 5 nodes, while the other two are each shared by 4 nodes).

We have run a number of simulations to determine the frame size and mean packet delay in these networks running HiPeR-ℓ for various values of the look-ahead ℓ. In our simulations we assume that the propagation delay $\tau = 20$ slots, the processing time $\nu = 100$ slots, and the tuning latency $\Delta = 4$ slots. Figures 4, 5, and 6 plot the actual and mean frame size when the look-ahead ℓ is 1, 2, and 3, respectively. The size of the first 3000 frames in the simulation is plotted. We can see that the mean is well-defined, and that the size of individual frames oscillates around this mean, as expected.

When ℓ is increased from 1 to 2, there is a significant decrease in the frame size. This can be explained by noting that, when $\ell = 1$, there is an idle period after the end of each frame equal to $\tau + \nu = 120$ slots (refer also to Figure 2). During this period, packets may arrive to the network nodes, but no packets are transmitted. Thus, the average frame size $\bar{F}$ has to be large enough so that, on average, the number of packets transmitted during $\bar{F}$ slots equals the number of packets arriving during $\bar{F} + 120$ slots. When $\ell = 2$, the propagation delay and processing time of 120 slots are completely overlapped with the transmission of the next frame, no idling occurs, and the frame size is smaller. Since $\ell = 2$ is sufficient to completely mask the 120 slots of propagation delay and processing time, there is nothing to gain from making $\ell = 3$, and the average frame size is not affected.

In Figure 7 we show the delay versus throughput curves for this network. The mean delay values are plotted with 95% confidence intervals, which, however, are so narrow that they are not visible. A look-ahead of 1 has the worst performance, as expected. Also, at low loads, $\ell = 3$ provides for shorter delays than $\ell = 2$, while the opposite is true for higher loads. At low loads, few packets arrive during a frame, thus the average frame size when $\ell = 2$ is not large enough to completely overlap the propagation delay and processing time. Thus, idling occurs after the end of two frames, and the result is longer delays than a look-ahead $\ell = 3$. As the load increases, the average frame size for $\ell = 2$ also increases. When the load is such that the 120 slots are completely masked with $\ell = 2$, no further gain is possible by using $\ell = 3$. That is, a look-ahead $\ell = 3$ will not decrease the frame size, but will increase the delay, as seen from (12). Finally, when $\ell = 4$ or more there is no advantage compared to $\ell = 3$, resulting in a higher delay.

7 Concluding Remarks

We have considered the media access problem arising in single-hop WDM networks. We introduced HiPeR-ℓ, a new reservation protocol designed to overcome the problems posed by non-negligible processing, tuning, and propagation delays. In HiPeR-ℓ, nodes send multiple reservation requests in a single control packet. As a result, the control requirements of the protocol are low, and nodes can use algorithms that schedule multiple packet transmissions on each wavelength, effectively masking the tuning latency. The parameter ℓ controls the degree of pipelining in the operation of the protocol, and can be used to mask the propagation delay and the processing time. We have derived a condition for the protocol to reach stability that mathematically captures the effect of load balancing and of the efficiency of the scheduling algorithm on the the overall network performance.

References

[1] M. Azizoglu, R. A. Barry, and A. Mokhtar. Impact of tuning delay on the performance of banwidth-limited optical broadcast networks with uniform traffic. *IEEE JSAC*, 14(5):935–944, June 1996.

[2] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Inc., 1992.

[3] M. S. Borella and B. Mukherjee. Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies. *IEEE JSAC*, 14(5):923–934, June 1996.

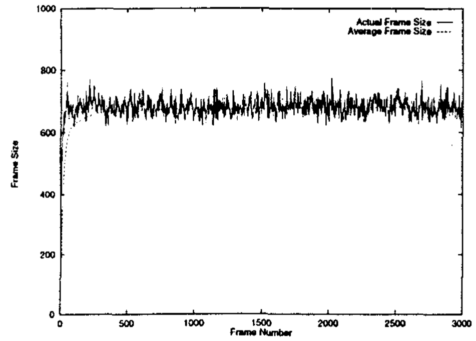

Figure 4: Frame size when the look-ahead is $\ell = 1$

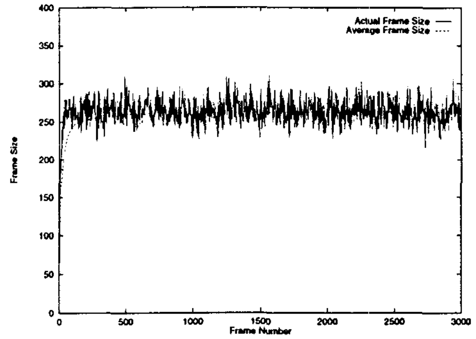

Figure 5: Frame size when the look-ahead is $\ell = 2$

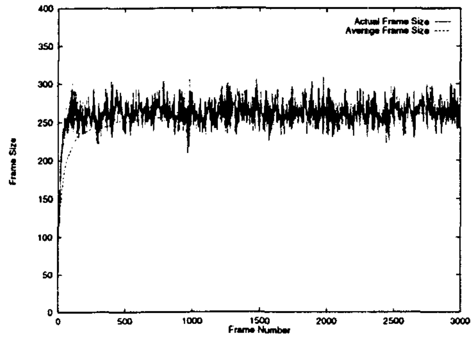

Figure 6: Frame size when the look-ahead is $\ell = 3$

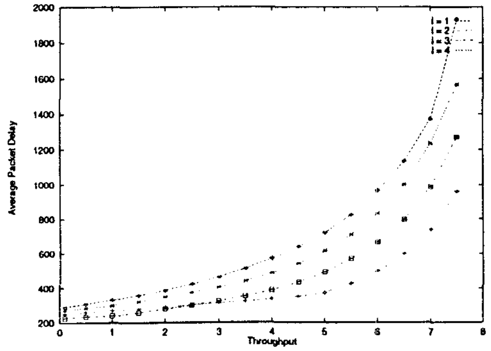

Figure 7: Delay vs. throughput for the client-server network

[4] Mon-Song Chen, N. R. Dono, and R. Ramaswami. A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks. *IEEE JSAC*, 8(6):1048-1057, August 1990.

[5] E. M. Foo and T. G. Robertazzi. A distributed global queue transmission strategy for a WDM optical fiber network. *INFOCOM '95*, pages 154-161. April 1995.

[6] P. A. Humblet, R. Ramaswami, and K. N. Sivarajan. An efficient communication protocol for high-speed packet-switched multichannel networks. *IEEE JSAC*, 11(4):568-578, May 1993.

[7] D. A. Levine and I. F. Akyildiz. PROTON: A media access control protocol for optical networks with star topology. *IEEE/ACM Trans. Networking*, 3(2):158-168, April 1995.

[8] A. Muir and J. J. Garcia-Luna-Aceves. Distributed queue packet scheduling algorithms for WDM-based networks. *INFOCOM '96*, pages 938-945. March 1996.

[9] B. Mukherjee. WDM-Based local lightwave networks Part I: Single-hop systems. *IEEE Network Magazine*, pages 12-27, May 1992.

[10] D. Park, H. G. Perros, and H. Yamashita. Approximate analysis of discrete-time tandem queueing networks with bursty and correleated input traffic and customer loss. *Operations Research Letters*, 15:95-104, 1994.

[11] G. Pujolle and H. G. Perros. Queueing systems for modelling ATM networks. In *Int'l Conf. on the Performance of Distributed Systems and Integrated Comm. Networks*, pages 10-12, Kyoto, Japan, September 1991.

[12] G. N. Rouskas and V. Sivaraman. On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies. *INFOCOM '96*, pages 1217-1224. March 1996.

[13] I. Rubin and Z. Zhang. Message delay analysis for TDMA schemes using contiguous-slot assignments. *ICC '88*, pages 418-422, 1988.

[14] K. Sivalingam and P. Dowd. A multi-level WDM access protocol for an opticall interconnected multi-processor system. *IEEE/OSA JLT*, 13(11):2152-2167, Nov. 1995.

[15] K. Sivalingam and J. Wang. Media access protocols for WDM networks with on-line scheduling. *IEEE/OSA JLT*, 14(6):1278-1286, June 1996.

[16] S. Tridandapani, J. S. Meditch, and A. K. Somani. The MaTPi protocol: Masking tuning times through pipelining in WDM optical networks. *INFOCOM '94*, pages 1528-1535. June 1994.