Distributed Control Protocols For Wavelength Reservation And Their Performance Evaluation [1]

X. Yuan

Department of Computer Science

Florida State University

R. Melhem and R. Gupta

Department of Computer Science

The University of Pittsburgh

and

Y. Mei and C. Qiao

Departments of CSE and EE

University at Buffalo (SUNY)

ABSTRACT

This paper describes distributed (or decentralized) protocols for establishing wavelength paths in point-to-point WDM networks. Distributed control improves reliability as well as scalability and reduces implementation cost of a network, but also presents major challenges in managing/allocating wavelengths efficiently. Two types of distributed wavelength reservation protocols are proposed and evaluated, namely forward and backward. The benefit of wavelength conversion is assessed first based on the evaluation of the forward reservation protocols, and it is found that wavelength conversion can result in a performance advantage in a distributed environment. For networks without wavelength conversion, a class of backward reservation protocols is studied and shown to generally perform better than their forward counterparts.

[1]This work was supported in part by NSF award MIP 9409864 and ANIR 98-01778 to University at Buffalo (SUNY), and by NSF awards CCR-9157371 and MIP 96-33729 to the University of Pittsburgh.

1 Introduction

All-optical networks, especially *Wavelength–division multiplexed* (WDM) networks, have received an enormous amount of attention, as exemplified by several projects including AON [2], MONET [20], TBONE [8] and various European projects (e.g. *ACTS-OPEN*[4]). In such networks, a lightpath is usually established before data is transferred between two communicating nodes, thus eliminating the need for buffering and electronic-to-optical or optical-to-electronic conversions during the data transfer at the intermediate nodes. Most of these lightpaths may be semi-permanent when used to provide *virtual topologies* to higher layers (e.g., ATM or SONET/SDH) but some applications such as those supporting bursty traffic directly on top of WDM or fast protection and restoration at the WDM layer may require frequent reconfiguration of the lightpaths.

Two basic approaches, namely *path multiplexing* (PM) and *link multiplexing* (LM) [13, 16], can be used to establish lightpaths in all-optical WDM (or TDM) networks. In PM, a lightpath is established by using the same wavelength that is available on all the links along the path. In LM, different wavelengths that are available on different links along the path can be used provided that all-optical wavelength converters are available at each node [11, 24].

The network control (or signaling) required for establishing a connection may be either *centralized* or *distributed.* In large networks, centralized control is not feasible, and hence it is essential to study distributed control protocols. In this paper, we first describe distributed *forward wavelength reservation* protocols suitable for both PM and LM. In these protocols, a source node sends a *reservation* packet to its destination once it has a connection request. The packet is processed by each node along the path, and as a result, an appropriate wavelength on the next link is reserved for the connection. We compare the performance of these protocols in LM and PM, and show that LM results in a better performance than PM in terms of throughput and connection latency.

If wavelength converters are not available at intermediate nodes, then PM should be used. In order to improve the performance of PM reservations, we consider a class of *backward reservation* protocols which overcomes the main disadvantages of PM forward reservations. In these protocols, the source node sends a *probe* packet to the destination instead of a reservation packet. The probe packet collects the wavelength usage information along the path, and upon receiving it, the destination node starts reserving wavelengths along the reverse path towards the source node. When used in PM, a reservation protocol, whether forward or backward, may differ from each other in its *aggressiveness*. For instance, a *conservative* protocol may examine one wavelength at a time, while an aggressive one may examine *several* wavelengths at a time.

Establishing connections under distributed control in non-optical networks has been extensively studied (see [7] for example). Establishing lightpaths in WDM networks via wavelength assignment for which centralized control is implicitly assumed, has been also studied (see for example [1, 5, 18]). In [17], a distributed protocol that sets up and tears down lightpaths for ATM connections was proposed. The protocol, whose performance was not evaluated in [17], is similar to a forward conservative one for PM according to our classification. Our work is based on the studies of distributed control protocols for optical TDM networks [15, 22], and especially WDM optical networks [10, 14, 21]. It is also related to the studies which evaluate the effectiveness of LM (i.e. having wavelength converters) assuming either static communications in [3, 12, 13], or dynamic communications under centralized control in [9, 16, 19, 23].

The rest of the paper is organized as follows. Section 2 outlines the general network control and protocol architectures. Section 3 presents the forward reservation protocols for PM and LM, and compares their performance. Section 4 presents backward reservation protocols for PM and evaluates their performance. Section 5 concludes the paper.

2 Network Control and Protocol Architectures

Figure 1 illustrates an example network control architecture. Each host is attached to a router consisting of an electronic control unit (CU) and an optical switch whose state is controlled by the CU. The latter can be a wavelength sensitive cross-connect or a wavelength-interchanging cross-connect [20] in PM and in LM, respectively. The optical switches are interconnected through *data channels* drawn as bold lines, while the control units are interconnected through *control channels* drawn as dashed lines.

The *data network*, consisting of the optical switches and the data channels, is used to form lightpaths, and operates in circuit-switching mode. On the other hand, the control network, consisting of the CUs units and the control channels, is used to exchange control information and operates in packet-switching mode.

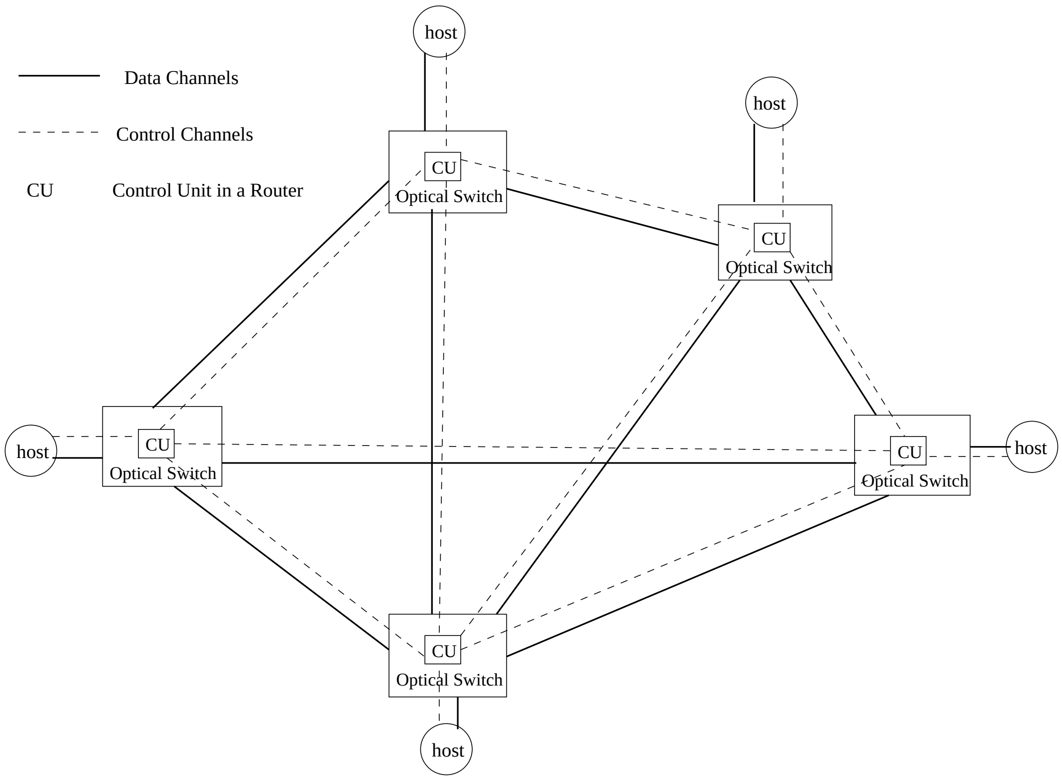

Figure 1: An example of an optical network with distributed control

The traffic in the control network consists of small control packets, and thus is much lighter than the traffic in the data network. Accordingly, the control channels can be implemented by one or more dedicated wavelengths in the same fiber link, or by a separate fiber link. Alternatively, electronic interconnects may be used for control channels as well. In the following discussions, the terms "link" will refer to a connection between two nodes on either the data or the control network. No direction will be associated with a link. A "channel" on a link will refer to a directional channel on the data network. Given that one "wavelength" is used per channel, we will usually use "wavelength" and "channel" interchangeably.

In addition to setting the optical switches, the CU at a router maintains the status of every wavelength on every link emerging from that router. A wavelength λ on a given link can be in one of the following three states:

- $AVAIL$: indicates that the wavelength λ is available on that link and can be used to establish a new connection,
- $LOCK$: indicates that λ is reserved for a possible use by some connection.
- $BUSY$: indicates that λ is being used by some connection.

Note that at the time a wavelength on a link is reserved, it may not be known if the connection can be successfully established using this wavelength or not. Accordingly, the $LOCK$ and $BUSY$ states are useful for distinguishing two phases in the reservation process, namely the phase prior to and the phase after making the commitment to using the wavelength for the connection, respectively. The two states can be combined into one state but are kept separate to clarify our presentation. If a wavelength is in either the $LOCK$ or the $BUSY$ state, the CU also maintains a field, namely *connection id* (or cid), which uniquely specifies the connection that is locking or is using the wavelength. Obviously, other connections can not reserve this wavelength. Instead, they can only reserve wavelengths that are in the $AVAIL$ state. For a link l, the set of wavelengths in state $AVAIL$ is denoted by $Avail(l)$. This set may be implemented by a bit vector of length W, where W is the total number of wavelengths, and bit i, $0 \leq i \leq W-1$, is 1 if wavelength i is in $Avial(l)$ and is 0 otherwise.

A wavelength reservation protocol may be characterized according to its *aggressiveness*. Specifically, in an *aggressive* reservation, the protocol tries to establish a connection by locking as many wavelengths on each link as possible first, hoping one of them can be used to establish the connection later. In a *conservative* reservation, the protocol locks only one wavelength on each link at a time during the reservation process.

In the design and evaluation of distributed control protocols, one needs to consider not only the effect of control overhead, but also *deadlock prevention*. *Deadlock* may occur in the control network due to one of the following reasons:

• Contention for wavelengths among control packets. Specifically, deadlock can occur if a request is allowed to lock wavelengths (forever) on some links while requesting wavelengths on other links. This type of deadlock can be avoided by following either a *dropping* or a *holding* policy, which characterizes the persistence of a protocol when the reservation for a connection cannot proceed because there is no appropriate wavelength on the next link. Specifically, under the dropping policy, the protocol immediately releases the wavelengths locked on the partially established path, while under the holding policy, the protocol keeps the wavelengths on the partial path locked for some period of time, hoping that during this period, the reservation will progress. If the reservation does not progress at the end of the period, then the request is dropped, and locked wavelengths are released.

• Lack of buffer space at a router, which will prevent control packets from moving forward on the control network, irrespective of the availability of wavelengths. This type of deadlock can be avoided by careful route and buffer management [6]. In our modeling and simulations, for example, we assume that every control packet with a given source-destination pair has a pre-determined route. In addition, since each control packet is small (e.g. a few bytes), and the traffic in the control network is light, it is possible to make the buffer at each router large enough to accommodate all control packets that can possibly arrive at that router.

• Loss of control information due to transmission errors or link/node failures. In both cases, deadlock can occur because all wavelengths may end up in the state of *LOCK* or *BUSY* but cannot be changed to *AVAIL*. A reliable transport (signaling) protocol for control packets is sufficient to avoid deadlock that may result from transmission errors. As for link/node failures, assuming that the network is survivable, deadlock can be avoided by associating a timer (or timers) with every wavelength in the state of *LOCK* or *BUSY*. Specifically, if the state of a wavelength is not changed from *LOCK* or *BUSY* to *AVAIL* within a certain time-out period, and no signal is detected during the entire period (heart-beat signals should be used to avoid labeling non-faulty nodes as faulty), then the state of that wavelength is automatically changed to *AVAIL*. Note that, the time-out period should be large enough to prevent pre-mature abortion of an on-going reservation.

We will not discuss the issue of deadlock avoidance any further in this paper but instead, concentrate on other aspects of the various reservation protocols.

3 Comparing PM and LM Under Distributed Control

3.1 Forward Aggressive Reservation in PM

In forward reservation protocols, there are four types of control packets, and every control packet related to a connection has a field, *packet.cid*, which carries the identifier of the connection. A unique connection identifier can be chosen at a particular source node by concatenating the address of the source node with a unique serial number identifying the connections originating at this source node. Two control packets are said to correspond to each other if they have the same value in the field *cid*. The four types of control packets used in an aggressive forward reservation protocol in PM are as follows:

• *Reservation packets* (RES). A source node sends a RES packet to reserve wavelengths. In addition to $RES.cid$, a RES packet contains a field $RES.wave\_set$ to keep track of the set of wavelengths that may be used to establish a connection. The wavelengths in $wave\_set$ are locked (i.e, their states are changed from $AVAIL$ to $LOCK$) at intermediate nodes while the RES packet progresses toward its destination node.

• *Acknowledgment packets* (ACK). A destination node sends an ACK packet to inform the source node of the success of connection establishment. The ACK packet follows the reverse of the path taken by the corresponding RES packet, and contains a $ACK.channel$ field which indicates the wavelength selected for the connection. It changes the state of the wavelength selected for the connection (from $LOCK$) to $BUSY$, and unlocks all other wavelengths previously locked by the corresponding RES packet (i.e. changes their states from $LOCK$ to $AVAIL$). The optical switches along the path are also set properly.

• *Negative Ack packets* (NAK). An intermediate node sends a NAK packet to inform the source node of the failure of its connection request. It follows the reverse of the path taken by the corresponding RES packet, and unlocks all wavelengths that were previously locked by the corresponding RES packet.

• *Release packets* (REL). A source node sends a REL packet to its destination node to release an established connection. It follows the path taken by the corresponding RES packet, and changes the state of the wavelength reserved for that connection from $BUSY$ to $AVAIL$.

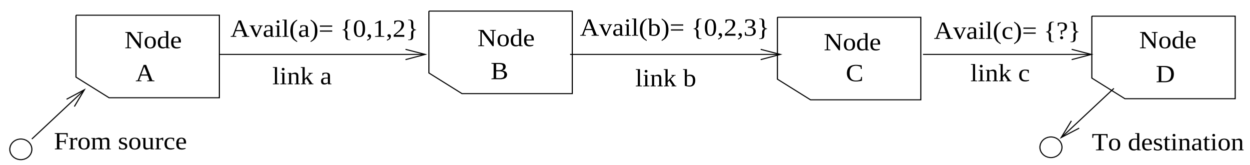

Figure 2: Establishing a connection under distributed control

We will use Figure 2 as an example to illustrate how the *forward aggressive reservation protocol with dropping* (or FAD) works in PM. When the source node wishes to establish a connection, it composes a RES packet with $RES.wave\_set$ set initially to include all W wavelengths, or in other words, $RES.wave\_set = \{0, 1, ..., W-1\}$. The RES packet is then routed to the destination through node A. Node A determines the next outgoing link, a, for the RES packet and updates $RES.wave\_set$ to $RES.wave\_set \cap Avail(a) = \{0, 1, 2\}$. These three wavelengths on link a are then locked. When intermediate node B receives the RES packet, it determines the next outgoing link b for the RES packet, and updates $RES.wave\_set$ to $RES.wave\_set \cap Avail(b) = \{0, 2\}$. Accordingly, wavelengths 0 and 2 on link b are locked. The RES packet is then forwarded to the next node, C, which determines the next outgoing link, c and updates $RES.wave\_set$ to $RES.wave\_set \cap Avail(c)$. Note that if both $RES.wave\_set$ and $Avail(l)$ for each link l are represented by a bit-vector, then $RES.wave\_set \cap Avail(l)$ is simply a bit-wise AND operation.

If $Avail(c)$ does not include wavelengths 0 or 2, then the resulting $RES.wave\_set$ is the empty set, ϕ. In this case, the connection cannot be established, and a NAK packet is sent back to the source node through nodes B and A. However, if $Avail(c)$ includes 0 and/or 2, then the resulting $RES.wave\_set$ is not empty and the wavelength(s) in $RES.wave\_set$ on link c will be locked. In general, the path is reserved incrementally as RES is routed to the destination. Once RES reaches the destination with a non-empty $RES.wave\_set$, the destination selects from $RES.wave\_set$ a wavelength, say λ, which is to be used for the connection. It then informs the source node (as well as the intermediate nodes, C, B and A) by sending an ACK message with $ACK.channel$ set to λ. The source can start sending data once it receives the ACK packet. After all data is sent, the source node sends a REL packet to tear down the connection.

3.2 Variations in Forward Reservation

The FAD protocol described above can be modified as follows:

Holding: the holding policy can be used instead of the dropping policy, resulting in a so-called *forward aggressive reservation with holding* (or FAH) protocol. With the holding policy, each RES has a field $RES.timeout$ whose value is set initially to the lifetime of the RES packet by the source node, and decreased by the intermediate nodes. When a FAH protocol is used, and $RES.wave\_set \cap Avail(c) = \phi$, the RES packet is buffered until either a wavelength becomes available or time-out is reached. In the example shown in Figure 2, node C buffers the RES packet for a limited period of time until either wavelengths 0 or 2 becomes available, or the value in $RES.timeout$ becomes 0. In the former case, the RES packet can then continue its journey and in the latter case, the RES packet is removed from the buffer and a NAK packet is sent back to the source. Comparing to dropping, holding can reduce the control overhead involved in tearing down the partially set-up path and having the source node re-establish it again later. However, holding requires a more complex implementation and the bandwidth held during the holding period may be wasted.

Conservative reservation: The aggressiveness of the reservation is reflected in the maximum size of the wavelength set, $RES.wave\_set$, initially chosen by the source node. In aggressive reservation, the source node sets the maximum size of $RES.wave\_set$ to W. On the other hand, the conservative reservation sets that maximum size to 1 to allow only a single wavelength to be included in $RES.wave\_set$. Thus, aggressive reservation will be successful as long as there exists a wavelength that is available along all the links in the path while conservative reservation can be successful only when the specific wavelength in $RES.wave\_set$ is available along all the links in the path. The aggressive reservation seems to increase the chance for a reservation to be successful. However, it may result in overly locking the wavelengths. As in forward aggressive reservation, either the drop or the hold policy may be adopted in forward conservative reservation, and we call such protocols FCD or FCH.

Protocols for LM: In LM, each switch is assumed to be able to convert wavelengths, and hence any available wavelength on a link along the path can be used to establish a connection. Accordingly, nothing can be gained by locking more than one wavelength during reservation.

In other words, aggressive protocols cannot have any advantage over conservative protocols, and thus only conservative protocols, such as FCD and FCH will be considered for LM.

3.3 Performance Evaluation

In this subsection, we use simulation to compare the performance of various forward reservation protocols for PM and LM. As a performance measure, we use the *throughput*, which is the amount of data transferred in a time unit. The effect of the following parameters on the average throughput are considered in this section:

• *request rate per node* R: It is assumed that requests are generated at each node according to a Poisson Process. In our simulations, R ranges from 0 to R_{max}, where R_{max} is the value that will saturate the network and is dependent of the protocol used as well as the values of the other parameters described below.

• *network size* N: The simulations are for two-dimensional meshes whose size ranges from 4×4 to 16×16 (i.e. $16 \leq N \leq 256$).

• *number of wavelength channels* W: This is the number of data channels per link.

The *average connection duration,* C, which represents the average amount of data to be transferred in each request, will be assumed to be 256 time units. Also, the *timeout value,* T, used in protocols that adopt the holding policy, will be assumed to be equal to 40 time units. We have experimented with other values for T and found that changing T does not have a significant effect on the performance of the protocols.

The following assumptions are made in the simulation: First, the destination of each requested connection is uniformly distributed. Second, when a source node receives an NAK packet, it will reinitiate the reservation process (e.g. by resubmitting a RES packet) after a random interval which has an exponential distribution with an average equal to the average connection duration. Third, the time needed to process a control packet at each node is the same, so is the time needed for a control packet to travel to the next node (they all have the default value of one time unit). Fourth, wavelength conversion in LM introduces a negligible delay. Finally, each node has an array of fixed transmitters and receivers that are capable of transmitting and receiving at different wavelengths simultaneously, and thus a node can have up to W connections active.

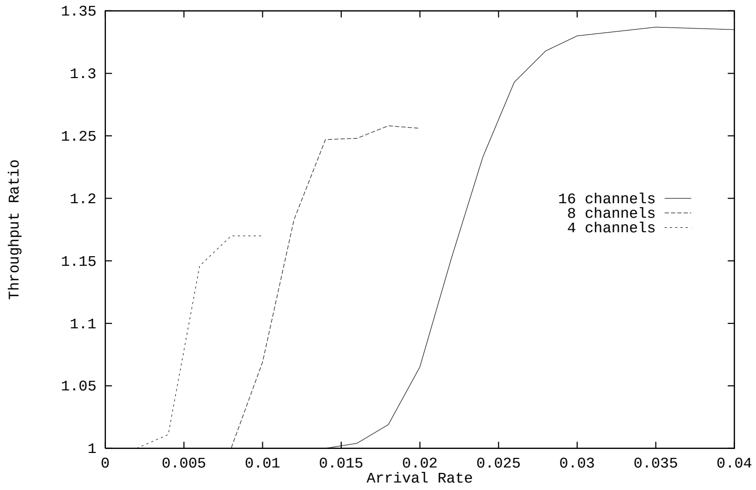

Figure 3: Effect of W on the ratio of LM-FCD and PM-FAD throughputs ($N = 64$, $C = 256$)

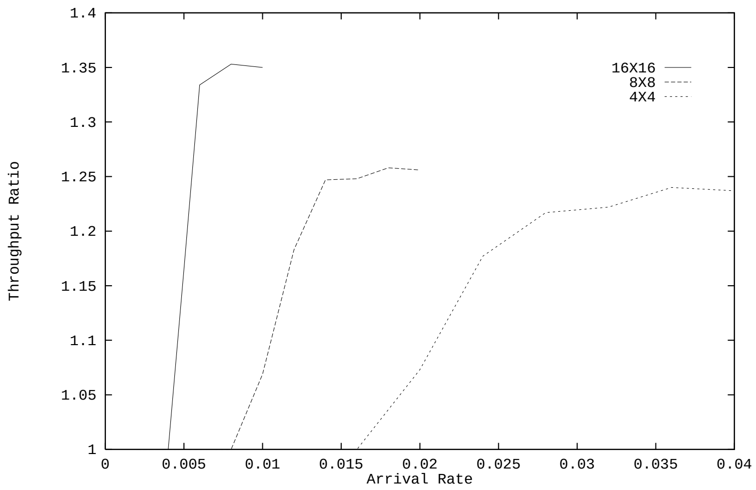

Figure 4: Effect of N on the ratio of LM-FCD and PM-FAD throughput ($W = 8$, $C = 256$)

Intuitively, in forward reservation, whether dropping is better than holding (in LM or PM), and whether aggressive is better than conservative (in PM only) depend on many parameters. In Figures 3 and 4 we compare the ratio of the highest throughput achieved by a forward reservation protocol in LM (which happens to be the throughput of LM-FCD), to the highest throughput achieved by a forward reservation protocol in PM (which happens to be the throughput of PM-FAD). As expected, the simulation confirmed that LM results in better performance (i.e. a higher throughput) than PM. Specifically, Figure 3 shows the effect of the number of wavelength channels on the throughput ratio. As can be seen, the throughput ratio increases with the number of wavelength channels. The reason is that with more wavelengths, LM has more choices at every intermediate node while PM has more choices only at the source node. Under distributed control, a failed attempt to reserve a wavelength in PM introduces a control penalty, which may be significant especially with a high failure rate under a medium to heavy traffic load. As can be seen from Figure 4, the effect of network size is similar in that the larger the network size, the bigger the performance advantage of LM over PM. This is because, when the network size increases, the average number of hops a connection has to go through is larger, and hence an attempt to reserve a wavelength is more likely to fail in PM.

4 Improving Protocol Performance in PM

If the nodes in a network do not have wavelength conversion capability, PM has to be used. In such a case, a forward aggressive protocol has the advantage that a connection request is guaranteed to succeed as long as a channel is available to establish that connection. The disadvantage, however, is that to obtain this guarantee, the protocol locks some channels that will end up not being used for the connection. On the other hand, a forward conservative protocol locks only one channel, but at the price of lowering the probability of being able to successfully establish a connection. The low probability is due to the fact that a channel for the connection has to be chosen on the first link without any knowledge about the availability of the channel on the remaining links. The *backward reservation* protocols presented in this section capitalize on the advantage of the aggressive scheme in being able to establish a connection whenever a free channel exists for that connection but without unnecessarily locking other channels. This is accomplished by first probing the network for the available channels in an aggressive way, and then making the actual reservation in a conservative way, as to be discussed next.

4.1 Backward Reservation

In contrast to the forward reservation, backward reservation uses a forward control packet to find the available channels in the network without locking them. A backward control packet then reserves one of the wavelengths found available. A backward reservation protocol requires five different types of control packets as discussed below. As before, every control packet carries a field *packet.cid*.

- *Probe packets* ($PROB$). A source sends a $PROB$ packet to its destination to gather information about wavelength usage without locking any wavelength. A $PROB$ packet carries a bit vector, $PROB.wave\_set$, to represent the set of wavelengths available to establish the connection.

- *Reservation packets* (RES). A destination node picks up one of the wavelengths in $PROB.wave\_set$ and sends a RES packet that follows the reverse of the path taken by the corresponding $PROB$ packet. The RES packet contains a $RES.channel$ field which indicates the wavelength chosen from $PROB.wave\_set$. An intermediate node receiving RES on a link, l, changes the state of $channel$ on l from $AVAIL$ to $BUSY$.

- *Fail packets* ($FAIL$). An intermediate node sends a $FAIL$ packet to inform the destination node of the reservation failure. The wavelength previously reserved by the corresponding RES packet is released (i.e. its status is changed from $BUSY$ to $AVAIL$)

- *Negative acknowledgment packets* (NAK). An intermediate node sends a NAK packet to inform the source node of the reservation failure. No special actions need to be taken by the intermediate nodes.

- *Release packets* (REL). The same as in the forward reservation. It is used to release connections.

As before, either the dropping or the holding policy may be adopted when making reservations. The backward reservation protocol with dropping works as follows. When the source node wishes to establish a connection, it composes a $PROB$ packet with $PROB.wave\_set$

containing all the wavelengths in the system. This packet is then routed to the destination. When an intermediate node receives the $PROB$ packet, it determines the next outgoing link, l, and updates $PROB.wave\_set$ to $PROB.wave\_set \cap Avail(l)$. If the resulting $PROB.wave\_set$ is empty, a NAK is sent back to the source node to indicate that the connection cannot be established. Fig. 5(a) shows this failed reservation case.

If the resulting $PROB.wave\_set$ is not empty, the node forwards $PROB$ to the next node. This way, as $PROB$ approaches the destination, the wavelengths available on the path are recorded in $PROB.wave\_set$. Once the destination node receives $PROB$, it composes a RES packet whose $RES.channel$ is set to one of the wavelengths in $PROB.wave\_set$, and sends it back to the source node. When an intermediate node receives the RES packet on a link, l, it checks if the wavelength $RES.channel$ is (indeed still) in $AVAIL(l)$, and if it is, it removes it from $AVAIL(l)$ and forwards RES to the next node toward the source node. However, it is possible that the state of $RES.channel$ has been changed to $BUSY$ since the $PROB$ packet recorded its availability. In this case, a NAK packet is sent to the source node to inform it of the failure, and a $FAIL$ packet is sent to the destination node to release the wavelength previously reserved by the corresponding RES. This process is shown in Fig. 5(b).

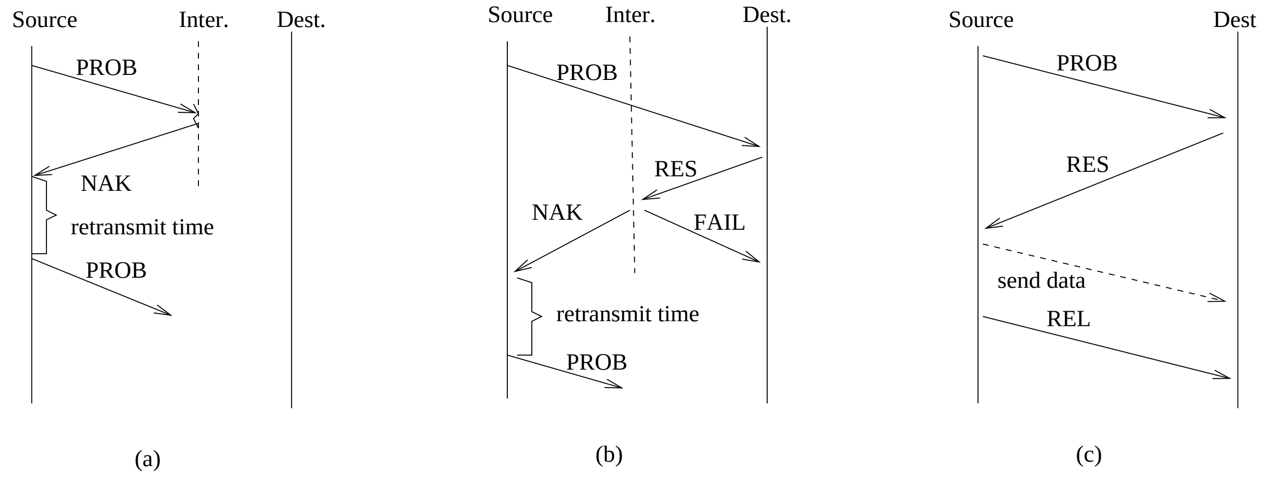

Figure 5: Control messages in backward reservation

When RES reaches the source node, that node can start sending data using the wavelength in $RES.channel$. After all data is sent, the source node sends a REL packet to tear down the connection. The process of successful reservation is shown in Fig. 5(c).

4.2 Performance Evaluation

In this section, we use simulation to compare forward and backward reservation protocols for PM. The parameters used in the simulation are the same as those described in Section 3.3. All other assumptions made previously are also kept the same. We will refer to the backward reservation protocols with holding and dropping by BCH and BCD, respectively, since as described earlier, RES follows a conservative approach for establishing a connection.

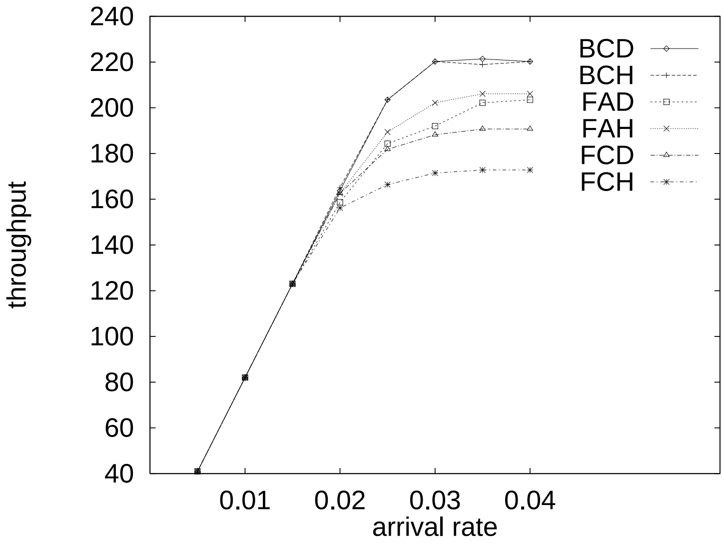

Figure 6: Throughput of reservation policies for PM (N = 64, W = 16, C = 128)
B=Backward, F=Forward, C=Conservative, A=Aggressive, D=Dropping, H=Holding

Figures 6 shows the throughput of different protocols on an 8×8 mesh assuming a connection duration of C=128. From the figure, it is clear that the backward reservation protocols outperform the forward counterparts. In the rest of this section, we will use the throughput at saturation (called maximal throughput) to compare the performance of different protocols.

In Figure 7, we show the relation between the maximum throughput and the connection duration assuming W = 32 and N = 64. It can be seen from this figure (and Figure 6 as well) that the performance of the backward reservation protocols is independent of whether "holding" or "dropping" is used. In other words, BCD is as good as BCH. For the forward conservative reservation, "dropping" has a better performance than "holding", (i.e. FCD is better than FCH), which is consistent with the results shown in the previous section.

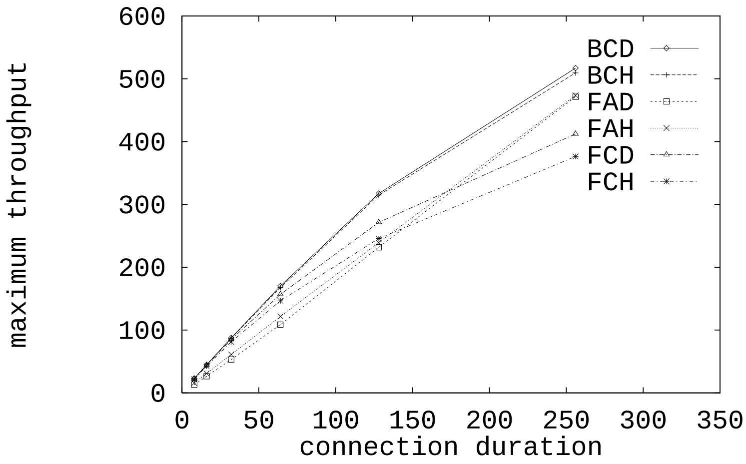

Figure 7: Effect of the connection duration on the maximum throughput ($N = 64$, $W = 32$)

However, for the forward aggressive reservation, "holding" has a slightly better performance than "dropping" (i.e. FAH is slightly better than FAD) when C is smaller than 256. This is because with FAH, the reservation can likely proceed after a short period of waiting as a result of one of the channels in $RES.wave\_set$ becoming available, whereas with FAD, the reservation will fail and subsequent retransmissions usually involve a relatively large overhead.

Figure 7 also shows that for a small C and a large W, forward conservative reservation is better than forward aggressive reservation. This is due to the fact that the number of additional channels that are locked unnecessarily in the aggressive reservation may be large due to a large W. In addition, the period during which these additional channels are locked in the aggressive reservation is long relative to the short average connection duration. Thus, the bandwidth wasted by the aggressive reservation becomes relatively large. In the remainder of this section, we will only compare the backward and forward conservative reservation protocols using "dropping".

In Figure 8, we plot the ratio between the maximum throughputs of BCD and FCD. This plot shows that the advantage of the backward protocol over the forward one depends on both the connection duration and the number of wavelengths. Specifically, for a given number of wavelengths, the advantage of the backward scheme increases when the connection duration

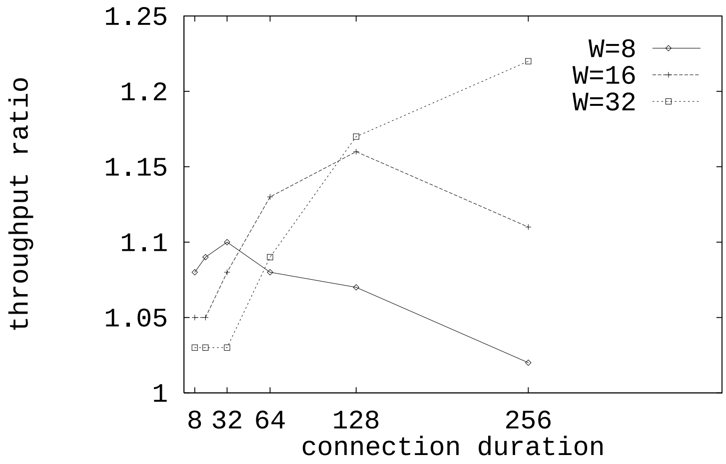

Figure 8: The ratio of BCD and FCD throughput for different C and W ($N = 64$)

increases up to a certain value, after which the advantage starts to decrease. For $W = 8$ and $W = 16$, the throughput ratio reaches its maximum at $C = 32$ and $C = 128$, respectively. For $W = 32$, the throughput ratio is expected to reache its maximum at much higher values of C. The larger the W, the more the throughput ratio is affected by C. Such a behavior can be explained as follows. First, with fewer wavelengths, the performance bottleneck is shifted from the performance of the reservation protocol (and the control network) to the bandwidth of the data network, thus affecting the throughput ratio. Similarly, for a given number of wavelengths, large connection durations means lower traffic on the control network, which also shifts the performance bottleneck away from the performance of the reservation protocol. At the other extreme, small connection durations means fast changes in the states of the channels, which render the information gathered by the PROB packet obsolete when it reaches the destination node, also affecting the backward scheme.

Finally, in Figure 9, we plot the ratio of the maximum throughput of the backward and forward protocols obtained by fixing the average connection duration at $C = 256$ and varying both the network size, N, and the number of wavelengths, W. These results show that the backward protocol outperforms the forward protocol (as evidenced by a ratio above 1) except for small network sizes (4×4). Increasing the number of channels, W, in general increases

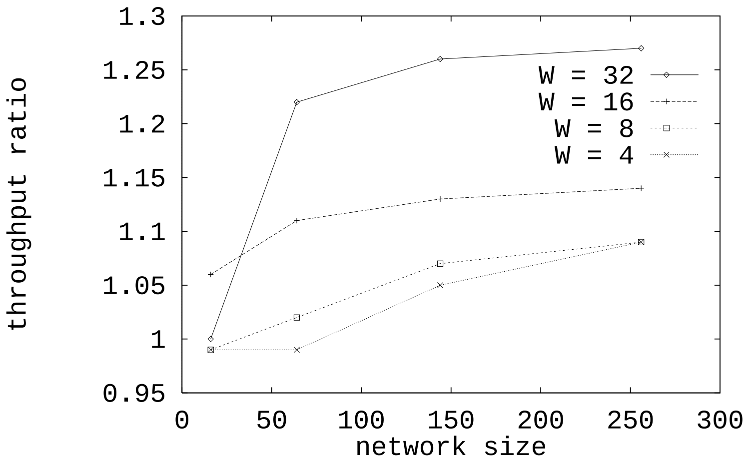

Figure 9: The ratio of BCD and FCD throughput for different N and W ($C = 256$)

the improvement in throughput obtained by the backward protocol. However, Figure 9 shows that for 4×4 networks, the throughput ratio decreases when W increases from 16 to 32. This indicates that, when W is excessively large relative to the network size, the number of channels available may exceed the communication requirement in the network, and thus the performance of the network will not depend on the control protocol used for reserving connections.

The results of this section about different reservation protocols for PM can be summarized as follows:

- When W is small, the choice of a particular protocol does not affect the performance (especially for high C and low N). The performance in this case is determined by the bandwidth of the data network.
- When W is excessively large relative to the network size (regardless how unlikely this may be), the choice of a particular protocol again does not affect the performance.
- Whenever the communication bottleneck is due to the control traffic, and connection durations are larger than the propagation time of the control packet, the backward scheme outperforms the forward scheme. The advantage of the backward scheme increases with

W due to the increase in the possible choices of wavelengths.

- For a given network size and number of channels, the advantage of the backward protocol reaches its maximum at a specific value of C. For lower values of C, the information about channel utilization gathered by the PROB packet becomes less accurate because the states of the channels change fast relative to the time taken to gather this information. For larger values of C, the communication bottleneck shifts from the efficiency of the reservation protocol to the bandwidth of the data network.

5 Conclusion

Establishing a path in WDM multi-hop networks under distributed control requires wavelength reservation protocols. In this paper, we have described various such protocols for WDM networks with and without wavelength conversions. In particular, for WDM networks without wavelength conversion, forward aggressive protocols are presented, which differ from conventional ones in that they attempt to make a reservation by locking more than one wavelength simultaneously. The main drawback of the forward aggressive protocols is that it may lock some channels that end up not being used. Backward reservation protocols are introduced to overcome this drawback.

Extensive simulations confirmed that the lack of wavelength conversion decreases the throughput of the network. Moreover, simulation also showed that a major portion of the throughput loss due to the absence of wavelength conversion may be regained if a backward reservation protocol is used instead of the forward protocol.

References

[1] A. Aggarwal et al. Efficient routing and scheduling in optical networks. *Proc. of the ACM-SIAM Symp. on discrete algorithms (SODA'93)*, 1993, pp. 412-423.

[2] S.B. Alexander et al. A precompetitive consortium on wide-band all-optical networks. *IEEE/OSA Journal of Lightwave Technology*, vol. 11, no. 5/6, 1993, pp. 714-735,

[3] R.A. Barry and P.A. Humblet. On the number of wavelengths and switches in all-optical networks. *IEEE Trans. on Communications*, vol. 42, 1994, pp. 583–591.

[4] M.W. Chbat. The optical Pan-European network (ACTS Project OPEN). In *Digest of IEEE/LEOS Summer Topic Meetings on Broadband Optical Networks: enabling technologies and applications*, August 1996, pp. 63-64.

[5] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: an approach to high-bandwidth optical WANs. *IEEE Trans. on Communications*, vol. 40, July 1992, pp. 1171-1182.

[6] W.J. Dally and C.L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Trans. on Computers*, vol. 36, May 1987, pp. 547-553.

[7] A. Girard. *Routing and dimensioning in circuit-switched networks.* Addison-Wesley, 1990.

[8] G.C. Hudek and D.J. Muder. Signaling analysis for a multi-switch all-optical network. In *Proceedings of Int'l Conf. on Communication (ICC)*, June 1995, pp. 1206-1210.

[9] M. Kovacevic and A. Acampora. On wavelength translation in all-optical networks. In *Proceedings of IEEE Infocom*, April 1995, pp. 413-422.

[10] Y. Mei and C. Qiao. Efficient distributed control protocols for WDM all-optical networks. In *Proceedings of Int'l Conf. on Computer Comm. and Networks (IC3N)*, Sept. 1997 pp. 150-153.

[11] X. Pan et. al. Dynamic operation of a three-port integrated Mach-Zehnder wavelength converter. *IEEE Photonic Technology Letters*, vol. 7, 1995, pp. 995-997.

[12] R. Pankaj and R.G. Gallager. Wavelength requirements of all-optical networks. *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, 1995, pp. 269-280.

[13] C. Qiao and Y. Mei. Off-line permutation embedding and scheduling in multiplexed optical networks with regular topologies. *IEEE/ACM Transactions on Networking*, April 1996, pp. 241-250.

[14] C. Qiao and Y. Mei. Wavelength reservation under distributed control. In *Digest of IEEE/LEOS Summer Topic Meetings on Broadband Optical Networks: enabling technologies and applications*, August 1996, pp. 45-46.

[15] C. Qiao and R. Melhem. Reconfiguration with time-division multiplexed MINs for multiprocessor communications. *IEEE Trans. on Par. and Dist. Sys.*, vol. 5, no. 4, 1994, pp. 337-352.

[16] C. Qiao and R. Melhem. Reducing communication latency with path multiplexing in optically interconnected multiprocessor systems. *IEEE Trans. on Par. and Dist. Sys.*, vol. 8, no. 2, 1997, pp. 97-108.

[17] R. Ramaswami and A. Segall. Distributed network control for wavelength routed optical networks. In *Proceedings of IEEE Infocom*, March 1996, pp. 138–147.

[18] R. Ramaswami and K.N. Sivarajan. Optimal routing and wavelength assignment in all-optical networks. In *Proceedings of IEEE Infocom*, June 1994, pp. 970–979.

[19] S. Subramaniam, M. Azizoglu, and A. Somani. Connectivity and sparse wavelength conversion in wavelength-routing networks. *IEEE Infocom*, March 1996, pp. 148–155.

[20] R. Wagner, R. Alferness, A. Saleh, and M. Goodman. MONET: Multiwavelength optical networking. *IEEE/OSA J. of Lightwave Tech.*, vol. 14, no. 6, 1996, pp. 1349-1355.

[21] X. Yuan, R. Gupta, and R. Melhem. Distributed control in optical WDM networks. In *IEEE MILCOM*, October 1996, pp. 100-104.

[22] X. Yuan, R. Melhem, and R. Gupta. Distributed path reservation algorithms for multiplexed all-optical interconnection networks. In *Proceedings of the IEEE Symp. High Performance Computer Architecture*, February 1997, pp. 48-57.

[23] J. Yates et. al. Limited-range wavelength translation in all-optical networks. In *Proceedings of IEEE Infocom*, March 1996, pp. 954-961.

[24] S. Yoo, C. Caneau, R. Bhat, and M. Koza. Transparent wavelength conversion by difference frequency generation in AlGaAs waveguides. In *Proc. Optical Fiber Communication Conference*, 1996, pp. 129-131.