Connection Management for Wavelength-Routed WDM Networks

Hui Zang†, Laxman Sahasrabuddhe, Jason P. Jue, S. Ramamurthy, and Biswanath Mukherjee
Department of Computer Science, University of California, Davis, CA 95616
{zang,sahasrab,juej,ramu,mukherje}@cs.ucdavis.edu
†Correspondence Author (Tel: (530) 752-5129; Fax: (530) 752-4767)

Abstract

In wavelength-routed WDM networks, a control mechanism is required to set up and take down all-optical connections. Upon the arrival of a connection request, this mechanism must be able to select a route, assign a wavelength to the connection, and configure the appropriate optical switches in the network. The mechanism must also be able to provide updates to reflect which wavelengths are currently being used on each link so that nodes may make informed routing decisions. In this work, we investigate and compare different distributed control mechanisms for establishing all-optical connections in a wavelength-routed WDM network.

1 Introduction

Wavelength-division multiplexing (WDM) has been rapidly gaining acceptance as a means to handle the ever-increasing bandwidth demands of network users [1]. All-optical WDM networks provide both huge bandwidth and end-to-end transparency by keeping the signal in the optical domain. In a wavelength routed WDM network, end users communicate with one another via all-optical WDM channels, which are referred to as *lightpaths*[1, 2]. A *lightpath* may span multiple fiber links. In the absence of wavelength converters, a lightpath must occupy the same wavelength on all fiber links through which it passes. Thus, given a set of lightpaths that need to be established on the network, and given a constraint on the number of wavelengths, we need to determine the routes over which these lightpaths should be set up and also determine the wavelength that should be assigned to these lightpaths. This problem is known as the *Routing and Wavelength Assignment* (RWA) problem. When setting up a lightpath, we need to configure various switches in the network. Typically, in connection-oriented networks, such as wavelength-routed WDM networks, a *network control and management protocol* is employed to configure the switches in the nodes through which the connection (i.e., the lightpath) is routed. In this paper we investigate two distributed control protocols: the link-state protocol proposed in [3] and a distributed routing protocol which is introduced in this paper. We evaluate the performance with regard to five metrics, namely, connection set up time, blocking probability, bandwidth requirement, stabilizing time and scalability. We assume that a connection utilizes only a single wavelength and that there is no wavelength conversion at intermediate nodes. Also, all control messages are transmitted on a *control layer*, i.e., a separate wavelength dedicated for control and management messages.

The rest of the paper is organized as follows. In Section 2, we describe two approaches for setting up and taking down lightpaths and the metrics to evaluate the performance of the approaches. In section 3 we analyze the performance of the approaches with regard to some of the metrics. We present numerical examples in Section 4. Section 5 concludes the paper and discusses areas for future research.

2 Protocol Descriptions

2.1 Link-state Approach

An approach for managing connections in a wavelength-routed WDM network is presented in [3]. In this approach, each node maintains the complete network topology, including the information about wavelengths that are in use on each link. Upon the arrival of a connection request, a node utilizes the topology information to select a route and a wavelength. Once the route and wavelength are selected, the node attempts to reserve the selected wavelength along each link in the route by sending reservation requests to each node in the route. If an intermediate node is able to reserve the wavelength on the appropriate link, it sends an acknowledgement directly back to the source node. If all of the reservations are successful, the source sends a SETUP message to each of the nodes. The appropriate switches are then configured at each node, and the connection is established. If even one of the reservations is not successful, then the call is blocked and the source node sends a TAKEDOWN message to each node in the route in order to release the reserved resources. When a connection is established or torn down, each node involved in the connection broadcasts a topology update message which indicates any changes in the status of wavelengths being used on the node's outgoing links. For further details of the protocol, including routing table characteristics and the update mechanism, we refer the reader to [3].

An important consideration in this protocol is the ef-

0-7803-5796-5/99/$10.00 © 1999 IEEE

fect of propagation delays on the connection set up time and topology updates. If two connections are being established nearly simultaneously, there may exist a race condition in which both connections are blocked even though it may be possible to establish one of the connections. Also, if a node attempts to set up a connection, but it hasn't received the latest topology information, then the node may attempt to reserve a wavelength on a link which is already in use, and the call will be blocked.

2.2 Distributed Routing Approach

In this work, we present a new connection set-up mechanism in which routes are selected in a distributed fashion without knowledge of the overall network topology. Each node maintains a routing table for each wavelength which specifies the next hop and the cost associated with the shortest path to each destination on this wavelength. The cost may reflect hop counts or link distances. The routing table is established by employing a distributed Bellman-Ford algorithm [4]. In the distributed routing approach, upon receiving a connection request, a node will choose the wavelength which results in the lowest cost (hop counts or link distance) to the destination, and it will forward the connection request to the next node in the path. Thus, the connection request is routed one hop at a time, with each node along the route independently selecting the next hop based on routing information, and reserving the appropriate wavelength on the selected link. Once the request reaches the destination node, the destination node sends an acknowledgement back to the source node along the reverse path (i.e. it sends the acknowledgement back to the node from which it received the connection request). Upon receiving the ACK, each node along the reverse path, configures its switch. The source node begins transmitting data after it receives the acknowledgement. If a node along the path is unable to reserve the desired wavelength on a link, it will send a negative acknowledgement back to the source along the reverse path. The nodes on the reverse path will release the reserved wavelengths as they receive the negative acknowledgement. The source node may then re-attempt the connection on a different wavelength. If the source node is unable to establish the connection on any wavelength, the call is blocked.

Once a connection is established, each node along the route sends to each of its neighbors an update message reflecting the status of the newly occupied link and wavelength. Each node receiving an update message may then update their routing tables, and send an update message to its neighbors if its routing tables are changed. Same thing happens when a connection is taken down.

The distributed routing approach has the following properties:

- Property 1: After a connection is set up, suppose no other connection requests arrive and no node or link failures occur in the network, then the routing tables in the network will stabilize within $2 \times DI$ iterations, where DI is the diameter of the network (in number of hops).
- Property 2: The protocol is free of deadlock, i.e., if a node sends a connection request, then it will receive a RESERVE-ACK or a RESERVE-NACK message within a fixed amount of time.
- Property 3: Suppose node A_1 sends a connection request to node A_n along the route A_1, A_2, ..., A_{j-1}, A_j, ..., A_{n-1}, A_n, and suppose the connection request reaches node A_j, then resources at nodes A_1, A_2, ..., A_{j-2}, A_{j-1} will be correctly reserved, i.e., all the resources along the route that are required to set up the connection have been reserved for this connection.
- Property 4: If the source node receives a RESERVE-ACK message, then all the resources along the route have been correctly reserved, i.e., all the resources along the route that are required to set up the connection have been reserved for this connection.

2.3 Comparison Metrics

We compare the two route-selection and connection set up approaches via simulation as well as analysis. The performance measures of interest are as follows:

- Connection set up time – time required to establish a connection once a connection request arrives.
- Blocking probability – probability that a connection cannot be established due to resource contention along the desired route.
- Stabilizing time – time required for nodes to update topology information after a connection has been established or taken down. If it takes a long time to update topology information, nodes may choose routes containing links that are already in use, leading to higher blocking.
- Bandwidth requirement for control messages – when a connection is set up or torn down, control information must be transmitted to the appropriate nodes. If updates need to be broadcast to all nodes, and updates occur frequently, then there may be significant loading on the control network.
- Scalability – how well the protocol handles the addition or deletion of nodes. Scalability is related to the bandwidth requirement and the amount of information stored at each node. The more bandwidth is required, the more information (network topology or routing tables) is stored at each node, the worse scalability the protocol has.

3 Analysis

We study the performance of the two approaches using the sample network shown in Figure 1 (network 1) and the one shown in Figure 2 (network 2). The numbers at the links represent the length of the links in number of 10-kilometers. Note that to make two approaches comparable, we have that, in the distribute routing approach, the source node does not attempt to set up a connection on a different wavelength when it receives a RESERVE-NACK - the call is blocked as in the link-state approach.

We assume the following:

- Message processing time at a node, P, is $10\mu s$.
- Time to configure, test and set up a cross-connect C, is $500\mu s$.

The following notations are need:

- Total number of nodes in the network N, $N = 6, 15$ for network 1 and network 2, respectively.
- Average propagation delay between two nodes D, $D = 0.10665ms$ for network 1 and $D = 1.82381ms$ for network 2.
- Average hop distance between two nodes H, $H = 1.533$ for network 1 and $H = 2.40952$ for network 2.

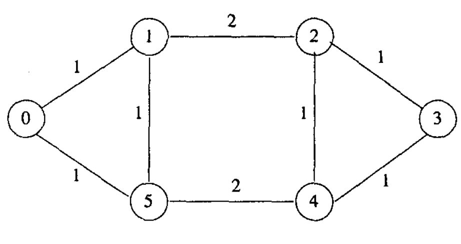

Figure 1: Network 1

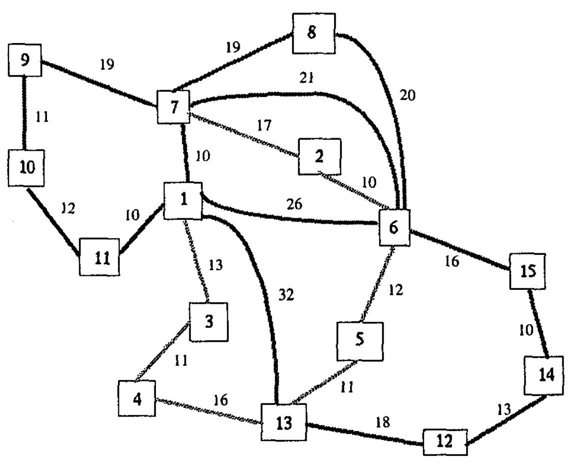

Figure 2: Network 2

3.1 Connection Set Up Time

Let us first calculate the connection set up time in the link-state approach. Two round trip propagation delays are need to set up the connection. Since the messages are sent to the nodes along the route simultaneously and processed by the nodes in parallel, $5P$ time units are needed to process the control messages before the switch is configured. The switches are then configured in parallel. Hence the time to set up a connection is:

$$T_L = 4D + 5P + C.$$

For the distributed routing approach, one round trip propagation delay is needed for the source to get the acknowledgement back after sending the request. Assume the nodes along the route are N_0, N_1, ..., N_H (N_0 is the source and N_H is the destination). N_0 processes 2 messages - the connection request and the RESERVE-ACK; N_1, ..., N_{H-1} each processes 2 messages - RESERVE and RESERVE-ACK; N_H processes only one message - RESERVE. As soon as each node receives a RESERVE-ACK (N_H receives RESERVE), it configures its switch. So switch configuration time is calculated only once. Hence the total time to set up a connection is:

$$T_D = 2D + (2H + 1)P + C.$$

We will see that the distributed routing approach gives shorter set up delays in both networks.

3.2 Bandwidth Requirement

There are two types of control messages delivered in the network:

1. signaling messages, such as RESERVE, TAKEDOWN, etc. Assume each message of this type has m_1 bytes in both approaches. Let B_1 denote the average bandwidth requirement of these messages for each connection to be set up and taken down.
2. network state update messages, such as routing tables. Assume each message of this type has m_{L2} bytes in the link-state approach and m_{D2} bytes in the distributed routing approach.

The number of type 1 messages for each connection be to set up or taken down is easy to calculate in both approaches. In the link-state approach, $5H$ control messages are required to set up a connection and H control messages are required to take down a connection. The bandwidth requirement is then:

$$B_{1L} = 6Hm_1.$$

In the distributed routing approach, on average, $3H$ control messages are sent for setting up a connection and H control messages are sent for taking down a connection. So,

$$B_{1D} = 4Hm_1.$$

If we assume that control messages are routed along the shortest path (not necessarily), then the number of type 2 messages that are sent in link-state approach is $2(N-1) \times H \times m_{L2}$. It is difficult to find an analytical expression for the number of control messages that are sent in the distributed routing approach. (This is a subject of ongoing research.) In this paper, only simulation results are presented for the number of control messages in the distributed routing approach.

3.3 Stabilizing Time

In the link-state approach, the stabilizing time is equal to the time it takes for a node's routing table to be delivered to the farthest node, which we denote as T_i for node i. T_i can be computed in this way. For each node j in the network and $j \neq i$, find the shortest route from i to j.

		setup time (ms)	bandwidth requirement (bytes / connecition)	stabilizing time (ms)
Network1	LS	0.9766	$9.2m_1 + 15.3m_{L2} = 4470.8$	0.17
	DR	0.754	$6.1m_1 + 86.6m_{D2} = 8856.3$	0.58
Network2	LS	7.845	$14.5m_1 + 67.2m_{L2} = 121192$	3.11
	DR	4.206	$9.9m_1 + 436.7m_{D2} = 104966$	7.92

Figure 3: Comparison of two approaches

Denote the number of hops on this route by H_{ij} and the propagation delay by D_{ij}. The time to transmit a routing table is R. Then the time it takes for node i's routing table to reach j is $H_{ij}R + D_{ij}$. We then find j such that $H_{ij}R + D_{ij}$ is maximum. So we have:

$$T_i = \max_j(H_{ij}R + D_{ij}).$$

Then the average stabilizing time will be $\frac{1}{N}\Sigma_i T_i$, where N is the number of nodes in the network.

We studied the average stabilizing time for the distributed routing approach through simulation, assuming that update messages in the Bellman-Ford algorithm are sent every $50\mu s$. The comparison is shown in Figure 3.

4 Numerical Results and Discussion

Figure 3 summarizes the results for the metrics that have been analyzed so far: LS stands for Link State Approach and DR stands for Distributed Routing Approach.

4.1 Connection Set Up Time

We observe that the distributed routing approach outperforms the link-state approach in terms of connection set up time. The distributed routing approach configures the switches as the acknowledgement is propagating back to the source, thus the connection set up time consists of one round-trip propagation delay between the source and destination. On the other hand, the link-state approach requires one round trip propagation delay to request and acknowledge the reservations, and another round trip propagation delay to configure the switches. Additional factors affecting the connection set up time are the processing delays at each node and the length of the route selected. The processing delays have more of an effect on the distributed routing approach, since connection set up messages are sent sequentially from one node to the next, while in the link-state approach, the connection set up messages may be sent in parallel to each of the nodes. With the current assumptions of nodal processing time and routing algorithm, the distributed routing approach performs better than the link-state approach.

4.2 Bandwidth Requirement

For a network with N nodes and W wavelengths, we have the following assumptions in our simulation:

- $m_1 = N + 1$;
- $m_{L2} = WN^2$, if each time W topology matrices are delivered and each topology matrix is of N^2 bytes; and
- $m_{D2} = 2WN$, if each time W routing tables are delivered and each table is of $2N$ bytes.

When a network grows larger, the length of type 2 messages grows faster in the link-state approach than in the distributed routing approach. Also, for a larger network, the update messages may not need to be broadcast to every node in the network each time there is a change in network state in the distributed routing approach, while in the link-state approach they have to. So as a network grows larger, the distributed routing approach performs better in terms of bandwidth requirement. As we can see from Figure 3, for simulation network 1 with 6 nodes, the link-state approach gives lower bandwidth requirement, while for simulation network 2 which has 15 nodes, the distributed routing approach gives lower bandwidth requirement. Note that in the results we presented in Figure 3, the number of type 2 messages in the distributed routing approach is obtained from simulation.

4.3 Stabilizing Time

With regard to stabilizing time, the link-state approach performs better for the two sample networks. This is due to the fact that the update information is broadcast only once for each update in network state, while in the distributed routing approach, a group of nodes may need to exchange information more than once to correctly set up the routing tables after each update. Also, we are studying the model in which each access station has one transmitter array and one receiver array. So each time a connection is set up at one wavelength between two nodes, the two nodes cannot accept other connection requests on the same wavelength. This information must be broadcast to every other node in the network. Under this assumption, in two cases of network state update - connection being set up and taken down, the stabilizing time in the distributed routing approach is at least as much as in the link-state approach. The bandwidth requirement in this model is also higher than it should be when multiple transceivers are used at each node.

To summarize, when two or more transceiver arrays are used at each access station and larger network are studied, we expect the distributed routing approach to give better performance for stabilizing time and bandwidth requirement than the link-state approach.

4.4 Blocking Probability

In Figure 4, we plot the blocking probability versus load for the two different connection management approaches using network 1 (Figure 1). It is similar for network 2. The results were obtained through simulation. The protocols were implemented in C++, with each protocol consisting of approximately 1500 lines of code.

We observe that the distributed routing approach yields lower blocking probabilities than the link-state approach, particularly at lower loads. This condition may partially be due to the difference in connection set up times between the two approaches. If the connection set up time is higher, there is a greater chance that multiple connection requests will be attempted simultaneously, resulting in a greater probability that the connections will block one another.

4.5 Scalability

For the network model and parameters considered in this work, the bandwidth required for the control messages in the link-state protocol is lower than that required in the distributed routing protocol for small networks. For larger networks (e.g. $N = 15$), the bandwidth required in the distributed routing protocol is smaller. For networks in which multiple transceivers are used at each node, we expect that the bandwidth required for control messages in the distributed routing protocol will be further smaller. We also noticed the fact that in the link-state protocol, information stored at each node is the entire network topology matrices on each wavelength, while in the distributed routing protocol, routing tables (each routing table consists of pairs of next-hop and cost for each destination node) on each wavelength are stored. So the amount of information stored at each node is in the order of WN^2 in the link-state protocol and in the order of WN in the distributed routing protocol. Hence, we expect that the distributed routing protocol is more scalable than the link-state protocol.

5 Conclusion

We introduced a distributed routing protocol for connection management in wavelength-routed WDM networks. We compared the distributed routing protocol with a link-state protocol for control management based on five metrics, namely, (a) connection set up time, (b) blocking probability, (c) stabilizing time, (d) bandwidth requirement for control messages, and (e) scalability. We simulated both protocol in two networks. For both networks, the link-state protocol has a lower stabilizing time than the distributed routing protocol, while the distributed

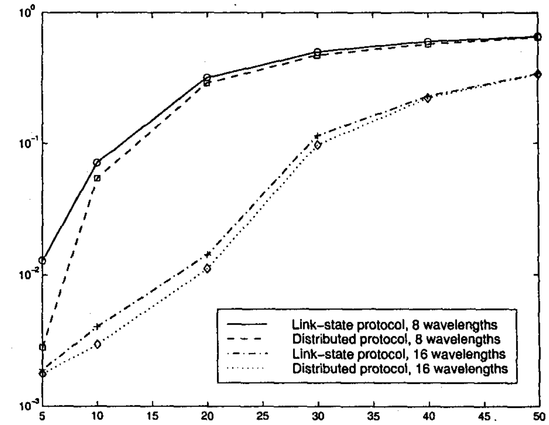

Figure 4: Blocking probability versus load for network 1 and 8 wavelengths.

routing protocol has a lower connection set up time than the link-state protocol. However, for larger networks and networks in which multiple transceivers are used at each access node, we expect that the stabilizing time in the distributed routing protocol to be lower. The link-state protocol has slightly higher blocking probability, which may be due to higher connection set up delays. The bandwidth required for control messages in the link-state protocol is lower for network 1 while higher for network 2 (which is larger) than that required in the distributed routing protocol. With lower bandwidth requirement for control messages and less amount of information stored at each node, the distributed routing protocol is more scalable than the link-state protocol.

Topics of ongoing investigation include the development of control protocol for setting up multicast connections and managing faults in the network.

References

[1] B. Mukherjee, *Optical Communication Networks,* McGraw-Hill, New York, 1997.

[2] R. Ramaswami, *Optical Networks,* Morgan Kaufmann, San Francisco, 1998.

[3] R. Ramaswami and A. Segall, "Distributed Network Control for Optical Networks," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6., pp. 936-943, Dec. 1997.

[4] J.J. Garcia-Luna-Aceves, "Distributed Routing with Labeled Distances," *Proc., IEEE Infocom '92,* Florence, Italy, May 1992.

[5] X. Zhang and C. Qiao, "Wavelength Assignment for Dynamic Traffic in Multi-fiber WDM Networks," *Proc., ICCCN '98,* Lafayette, Louisiana, USA.

[6] R. A. Barry and S. Subramaniam. "The MAX-SUM Wavelength Assignment Algorithm for WDM Ring Networks," *Proc., OFC '97,* Feb 1997.