Burst Segmentation: An Approach For Reducing Packet Loss In Optical Burst Switched Networks

Vinod M. Vokkarane, Jason P. Jue, and Sriranjani Sitaraman
Department of Computer Science, The University of Texas at Dallas
Richardson, TX 75083
vinod@utdallas.edu, jjue@utdallas.edu, ginss@utdallas.edu

***Abstract*– In this paper we address the issue of contention resolution in optical burst switched networks, and we introduce an approach for reducing packet losses which is based on the concept of burst segmentation. In burst segmentation, rather than dropping the entire burst during contention, the burst may be broken into multiple segments, and only the overlapping segments are dropped. The segmentation scheme is investigated in conjunction with a deflection scheme through simulation, and it is shown that segmentation with deflection can achieve a significantly reduced packet loss rate.**

I. Introduction

The amount of raw bandwidth available on fiber optic links has increased dramatically with advances in dense wavelength division multiplexing (DWDM). In order to efficiently utilize this bandwidth, an all-optical transport method, which avoids optical buffering while handling bursty traffic, and which supports fast resource provisioning and asynchronous transmission of variable sized packets, must be developed. Optical Burst Switching (OBS) is one such method for transporting traffic directly over a bufferless optical WDM network [1].

In optical burst switched networks, bursts of data consisting of multiple packets are switched through the network all-optically. A control message (or header) is transmitted ahead of the burst in order to configure the switches along the burst's route. The data burst follows the header without waiting for an acknowledgement for the connection establishment. The header and the data burst are separated at the source, as well as subsequent intermediate nodes, by an offset time, as shown in Fig. 1. The offset time allows for the header to be processed at each node while the burst is buffered electronically at the source; thus, no fiber delay lines are necessary at the intermediate nodes to delay the burst while the header is being processed. The control message may also specify the duration of the burst in order to let a node know when it may reconfigure its switch for the next burst, a technique known as *Delayed Reservation* (DR) [1]. In this paper, we will consider an optical burst switched network which uses the DR technique.

A major concern in optical burst switched networks is contention, which occurs when multiple bursts contend for the same link. Contention in an optical burst switched network is particularly aggravated by the highly variable burst sizes and the long burst durations. Furthermore, since bursts are switched in a cut-through mode rather than a store-and-forward mode, optical burst-switched networks generally have very limited buffering capabilities. While existing contention resolution schemes for photonic packet networks, such as deflection and buffering, may be utilized in optical burst switched networks, additional schemes may also be necessary in order to combat high contention rates and to achieve high network utilization.

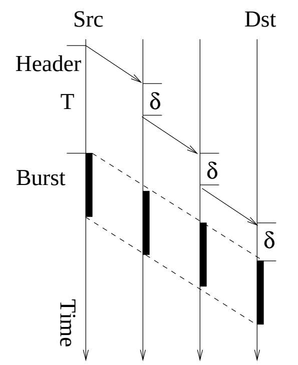

Fig. 1. The use of offset time in OBS.

In [2], an offset scheme was proposed for isolating classes of bursts, such that low-priority bursts do not cause contention losses for high-priority bursts; fixed and variable fiber delay line buffers were also utilized to further reduce blocking. In [2] and [3] contention is reduced by utilizing additional capacity in the form of multiple wavelengths. In both cases, optical wavelength conversion was assumed. While optical wavelength conversion has been demonstrated in laboratory environments, the technology is not yet mature, and the range of possible conversions is somewhat limited.

Most of the current literature deals with approaches to minimize burst losses rather than packet losses. In existing contention resolution schemes for optical burst switched networks, when contention between two bursts cannot be resolved through other means, one of the bursts will be dropped in its entirety, even though the overlap between the two bursts may be minimal. For certain applications, which have stringent delay requirements but relaxed packet loss requirements, it may be desirable to lose a few packets from a given burst rather than losing the entire burst. In this paper, we will introduce a new contention resolution technique called *burst segmentation*, in which only those packets which overlap with a contending burst will be dropped. The paper is organized as follows. Section II introduces the concept of burst segmentation and describes the segment dropping policies. Section III discusses

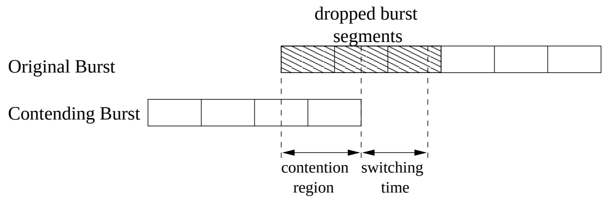

Fig. 2. Selective segment dropping for two contending bursts.

segmentation with deflection. Section IV compares the simulation results for different contention resolution policies in a specific network topology, and Section V concludes the paper.

II. Burst Segmentation

To overcome some of the limitations of optical burst switching, we introduce the concept of burst segmentation. The burst is divided into basic transport units called segments. Each of these segments may consist of a single packet or multiple packets, and the segments define the possible partitioning points of a burst when the burst is in the optical network. All segments in a burst are initially transmitted as a single burst unit. However, when contention occurs, only those segments of a given burst which overlap with segments of another burst will be dropped, as shown in Fig. 2. If switching time is non-negligible, then additional segments may be lost when the output port is switched from one burst to another.

There are two approaches for dropping burst segments when contention occurs between bursts. The first approach is to drop the tail of the first burst (Fig. 2), and the second approach is to drop the head of the contending burst. A significant advantage of dropping the tail segments of bursts rather than the head segments is that there is a better chance of in-sequence delivery of packets at the destination, assuming that dropped packets are retransmitted at a later time.

One issue that arises when the tail of a burst is dropped is that the header for the burst, which may be forwarded before the segmentation occurs, will still contain the original burst length; therefore, downstream nodes may not know that the burst has been truncated. If downstream nodes are unaware of a burst's truncation, then it is possible that the previously truncated tail segments will contend with other bursts, even though these tail segments have already been dropped at a previous node. These contentions may result in unnecessary packet loss.

If a tail-dropping policy is strictly maintained throughout the network, then the tail of the truncated burst will always have lower priority, and will never preempt segments of any other burst. However for the case in which tail dropping is not strictly maintained, some action must be taken to avoid unnecessary packet losses. A simple solution is to have the truncating node generate and send out a trailing control message to indicate when the truncated burst ends. In this policy, the offset between the trailer packet and the end of the truncated burst is similar to the offset between the header and the start of the burst.

In a head-dropping policy, the head segments of the contending burst will be dropped. A head-dropping policy will result in a greater likelihood that packets will arrive at their destination out of order. Also, the control message of the contending burst would need to be modified and delayed. The advantage of head-dropping is that it ensures that, once a burst arrives at a node without encountering contention, then the burst is guaranteed to complete its traversal of the node without preemption by later bursts.

In this paper, we consider a modified tail-dropping policy when determining which segment to drop. In this policy, the tail of the original burst is dropped only if the number of segments in the tail is less than the total number of segments in the contending burst. If the number of segments in the tail is greater than the number of segments in the contending burst, then the entire contending burst is dropped. This approach reduces the probability of a short burst preempting a longer burst and minimizes the number of packets lost during contention.

There are a number of additional issues and challenges which arise when implementing burst segmentation in practical systems:

1. Switching time: Since the system does not implement buffering or any other delay mechanism, the switching time is a direct measure of the number of packets lost during reconfiguring the switch due to contention. Hence, a slower switching time results in higher packet loss. While deciding which burst to segment, we consider the remaining length of the original burst, taking the switching time into account. By including switching time in burst length comparisons, we can achieve the optimal output burst lengths for a given switching time.

2. Segment boundary detection: In the optical network, segment boundaries of the burst are transparent to the intermediate nodes that switch the burst segments all-optically. At the network edge nodes, the burst is received and processed electronically. Since the burst is made up of many segments, the receiving node must be able to detect the start of each segment and identify whether or not the segment is intact. If each segment consists of an Ethernet frame, detection and synchronization can be performed using the preamble field in the Ethernet frame header, while errors and incomplete frames can be detected by using the CRC field in the Ethernet frame.

3. Trailer creation: The trailer has to be created electronically at the switch where the contention is being resolved. The time to create the trailer can be included in the header processing time, δ, at each node.

III. Segmentation with Deflection

A basic extension of burst segmentation is to implement segmentation with deflection. Rather than dropping segments of a burst, we can either deflect the entire burst or deflect segments of the burst to an output port other than the intended output port. This approach is referred to as deflection routing or hot-potato routing [4], [5]. Implementing segmentation with deflection (Fig. 3) increases the probability of the burst reach-

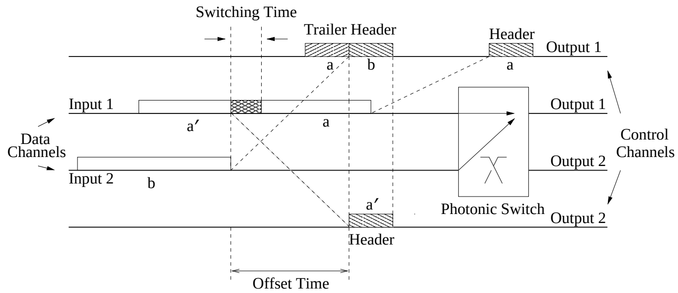

Fig. 3. Segmentation with deflection policy for two contending bursts.

ing the destination and hence improves the performance. One problem which may arise is that a burst may encounter looping or may be deflected multiple times, thereby wasting network bandwidth. In order to avoid these problems, when the hop-count of the burst reaches a threshold, the burst is dropped. This limitation on hop-count also ensures that the offset time maintains a reasonable value. Also, deflection increases blocking probability in high load conditions [6].

There could be one or many alternate deflection ports. The alternate deflection port(s) could be allotted ahead of time using fixed port assignment policy or using the second shortest path algorithm. A load balancing approach, which is based on the current link utilizations, could also be used so that the burst is deflected to an under-utilized link. In this paper, we consider only one alternate deflection port, and choose the port which results in the second shortest path to the destination.

Selection of which burst (or burst-segments) to deflect during contention could be done in one of the following ways. Firstly, the burst with shorter remaining length (taking switching time into account) could be deflected to the alternative port, or dropped if the alternate port is busy (Fig. 3). Secondly, we could incorporate priorities into the burst, so that in case of contention the lower priority burst is deflected or segmented based on the underlying policy.

Now combining segmentation with deflection, we have two approaches for ordering the contention resolution policies, namely, *segment-first* and *deflect-first*. In the segment-first policy, if the remaining length of the original burst is shorter than the contending burst, then the original burst is segmented and its tail is deflected, otherwise the contending burst is deflected. In case the alternate port is busy, the deflected part of the original burst is dropped. In the deflect-first policy, in case of contention, the contending burst is deflected if the alternate port is free. If the alternate port is busy and if the remaining length of the original burst is shorter, then the original burst is segmented and its tail is dropped. If the contending burst was found to be shorter, then the original burst is dropped. In this paper, we consider the segment-first policy.

An example of the segmentation-deflection scheme is shown in Fig. 3. Initially when the header for *burst a* arrives at the switch, it is routed onto output port 1. Once the header of the *burst b* arrives at the switch we have a contention. Since the offset time is common to all the bursts, the header indicates when and where the bursts will contend. So taking the switching time into consideration, and based on the segment-first policy, one of the bursts is deflected (or segmented and deflected) to the alternate port if it is free and is dropped otherwise. Here the remaining length of *burst a* is less than the length of *burst b*. Hence *burst a* is segmented and its tail is deflected to the alternate port as a new burst. A header is created for the deflected new burst and sent on the output 2 control channel. This new header generation is done at the time the header of *burst b* is processed. A trailer is created for the segmented *burst a* and is sent on the control channel of output 1. Packets of the burst to be segmented are lost during the reconfiguration of the switch. Hence faster switching time improves the performance. In the segmentation with deflection policy, the processing time δ (Fig. 1) at each node includes the time to create a header for the new burst segment in case of contention. Hence the offset time is same as in the case of standard optical burst switching.

A possible side-effect of segmentation with deflection is that, when there is contention, the shorter remaining burst will get segmented and will be deflected as a new burst. Creating these new short bursts may lead to burst fragmentation. The newly created short burst may contend with other bursts in the network, leading to additional fragmentation. Fragmentation is not a major issue, as the policies for deflection and dropping tend take care of the smaller burst. Every time a burst is segmented, the lengths of the two colliding bursts are compared and the smaller of the contending burst or the remaining part of the first burst is deflected or segmented respectively. Thus, the short, fragmented bursts will have lower priority and will not significantly hinder other bursts.

Another issue when implementing segmentation and deflection is how to handle long bursts which may span multiple nodes simultaneously. If a long burst passing through two or more switches experiences contention from two or more different bursts at different switches, then, based on the timing of these contentions, the contentions are resolved in the following manner:

If an upstream node segments the burst first, then the downstream nodes are updated by the trailer packet to eliminate unnecessary contentions. On the other hand, if the contention occurs at the downstream node before the upstream node, and if the burst's tail is deflected at the downstream node, then the upstream contentions will not be affected. If the downstream node drops the tail of the burst, then the upstream node will not know about the truncation and will continue to transmit the tail. The downstream node may send a control message to the upstream node in order to reduce unnecessary contentions with the tail at the upstream node.

IV. Simulation Results

In order to evaluate the performance of the segmentation and deflection schemes, we develop a simulation model. The following have been assumed to obtain the results:

- Burst arrivals to the network are Poisson with rate λ.
- Burst length is exponentially distributed with rate μ.
- Load is measured in Erlang.
- Transmission rate is 10 Gbps.
- Packet length is 1500 bytes.
- Switching time is 10 μs.
- There is no buffering or wavelength conversion at nodes.
- Each node handles both bypassing and locally generated or terminated bursts.
- Bursts are uniformly distributed over all sender-receiver pairs.
- Dijkstra shortest path routing algorithm is used to find the path between all node pairs.

Figire 4 shows the 14-node NSFNET on which the simulation was implemented. The distances shown are in km. We have compared four different policies for handling contention in the OBS network, they are:

- *Drop Policy (DP):* Drop the entire contending burst.
- *Deflect and Drop Policy (DDP):* Deflect the contending burst to the alternate port. If the port is busy, drop the burst.
- *Segment and Drop Policy (SDP):* Segment-first policy without deflection.
- *Segment, Deflect and Drop Policy (SDDP):* Segment-first policy with deflection.

Figure 5 plots the total packet loss probability versus the load for the four different contention resolution policies. An average burst length of $1/\mu$ = 100 ms is assumed. We observe that SDP performs better than DP in all load conditions, and the two policies with deflection namely, DDP and SDDP perform better than the corresponding policies without deflection at low loads. Also, at low loads DDP performs better than SDDP since there is no loss due to switching time in DDP; whereas, at high loads, SDDP is better than DDP. A logical explanation would be that, in segmentation, on average only half of the packets from one of the bursts are lost when contention occurs. Also, at low loads, there is a greater amount of spare capacity, increasing the chance of successful deflection. At high loads, deflection may add to the load, increasing the probability of contention, and thereby increasing loss.

Figure 6 shows the packet-loss performance at very high loads. SDDP performs the best when the load is under 50 Erlang, after which SDP performs better. DDP is good only at low loads, while at very high loads DP fares better than DDP. We observe that, at very high loads, policies without deflection perform better then the policies with deflection. This is because deflection increases the effective arrival rate within the network, which may lead to more contentions.

Figure 7 shows the average number of hops versus load. For the deflection policies, the number of deflections increase as the load increases, resulting in increasing average hop distance at low loads. As the load increases further, those bursts which are further from their destination will experience more contention than those bursts which are close to their destination. Thus,

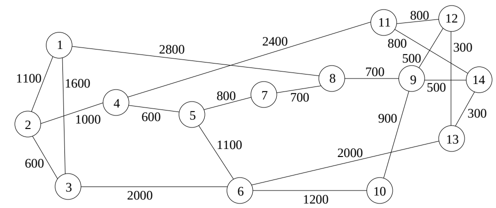

Fig. 4. Picture of NSFNET with 14 nodes.

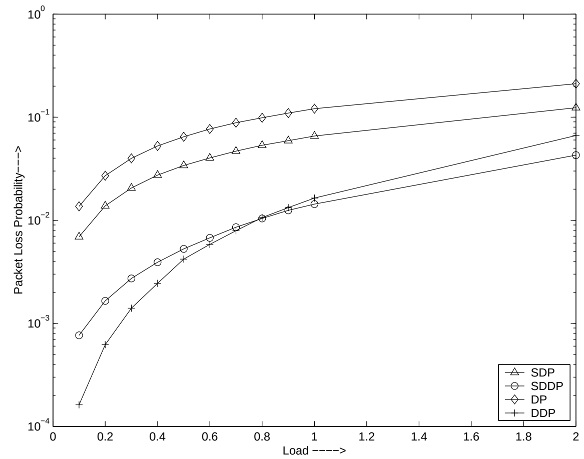

Fig. 5. Packet loss probability versus load for NSFNET at low loads with $\frac{1}{\mu} = 100$ ms.

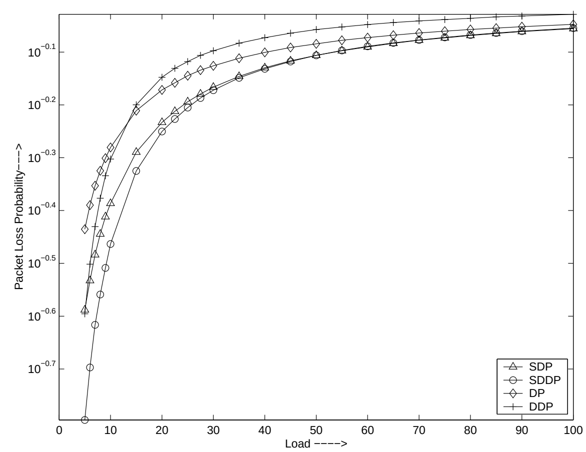

Fig. 6. Packet loss probability versus load for NSFNET at high loads with $\frac{1}{\mu} = 100$ ms.

bursts with higher average hop count are less likely to reach their intended destination, and the average hop distance will decrease as load increases.

Figure 8 shows the simulation results of the average output burst size versus the load for SDP and SDDP. The output burst size is measured over both dropped and successfully received bursts. Initially, the burst size decreases with increasing load, as there are more segmentations with the increasing number of contentions. As the load increases further, the segmented bursts encounter more contentions, and because the segmented bursts have smaller size (lower priority), they are dropped. The values

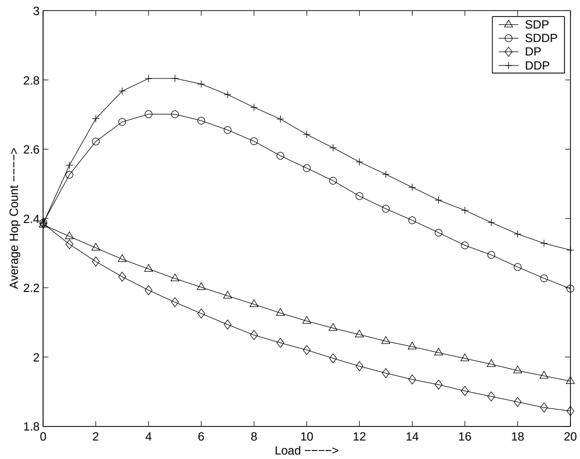

Fig. 7. Average number of hops versus load for NSFNET with $\frac{1}{\mu} = 100$ ms.

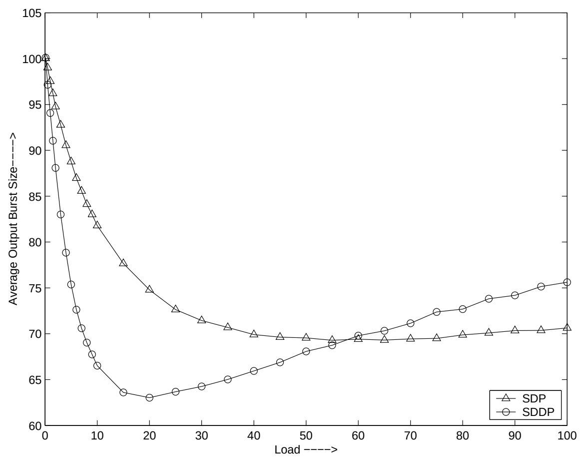

Fig. 8. Average output burst size versus load for NSFNET with $\frac{1}{\mu} = 100$ ms.

for DP and DDP are constant for different values of load as the size of the burst is never altered.

The packet loss probability versus load for different values of switching time is shown in Fig. 9. As the switching time increases, the performance of SDDP decreases as a greater number of packets are lost during the re-configuration of the switch. On the other hand, DDP is not affected by the switching time and is almost constant. At low switching times, the results show that SDDP is better than the standard DDP. While at higher switching times, the standard DDP is better than the new SDDP because of the loss of packets during the switching time.

V. Conclusion

In this paper we introduced the concept of burst segmentation for contention resolution in optical burst switched networks, and we investigated a number of different policies with and without segmentation and deflection. The segmentation policies perform better than the standard dropping policy, and

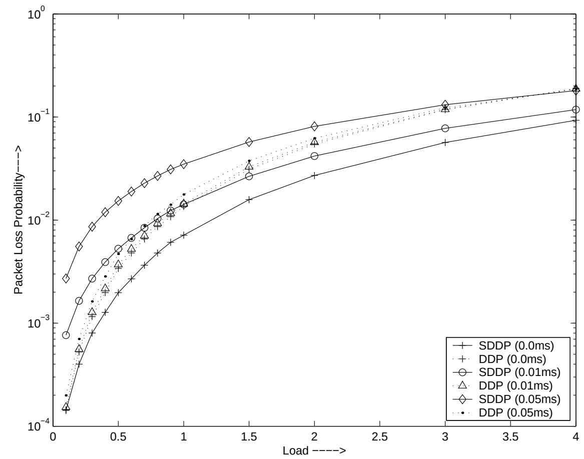

Fig. 9. Packet loss probability versus load at varying switching times for NSFNET with $\frac{1}{\mu} = 100$ ms.

offer the best performance at high loads. The policies which incorporate deflection tend to perform better at low loads.

An area for future work is the investigation of combined segmentation/deflection schemes in which deflection is performed before segmentation when a contention occurs. Also, in this paper, we considered only one alternate output port for deflection. Policies which consider multiple alternate output ports and in which the selection criteria is based on load and shortest path may also be considered. The segment dropping and deflection policies can also be implemented with priorities. Priorities would be based on a burst's tolerance for segmentation, deflection, and loss. To effectively evaluate the quality of service offered by various priority policies, a retransmission scheme for dropped packets could be implemented in order to measure end-to-end delay. A reasonable approach would be to implement a TCP layer on top of the optical burst switched layer. In such an implementation, it would also be useful to evaluate how TCP layer congestion control schemes react to and interact with various contention resolution schemes.

References

[1] C. Qiao and M. Yoo,"Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet,"*Journal of High Speed Networks*, vol. 8, no.1, pp. 69-84, Jan. 1999.

[2] M. Yoo, C. Qiao, and S. Dixit,"QoS Performance of Optical Burst Switching in IP-Over-WDM Networks,"*IEEE Journal on Selected Areas in Communications*, vol. 18, no.10, pp. 2062-2071, Oct. 2000.

[3] J.S.Turner, "Terabit Burst Switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3-16, 1999.

[4] A. S. Acampora and I. A. Shah, "Multihop Lightwave Networks: A Comparison of Store-and-Forward and Hot-Potato Routing," *IEEE Transactions on Communications*, vol. 40, no. 6, pp. 1082-1090, June 1992.

[5] A. Bononi, G. A. Castanon, and O. K. Tonguz, "Analysis of Hot-Potato Optical Networks with Wavelength Conversion, " *IEEE Journal of Lightwave Technology*, vol. 17, no. 4, pp. 525-534, April 1999.

[6] S. Yao, B. Mukherjee, S.J.B. Yoo, and S. Dixit,"All-Optical Packet-Switched Networks: A Study of Contention Resolution Schemes in an Irregular Mesh Network with Variable-Sized Packets," *Proceedings, SPIE OptiComm 2000*, Dallas, TX, pp.235-246, Oct. 2000.