Generalized Burst Assembly and Scheduling Techniques for QoS Support in Optical Burst-Switched Networks

Vinod M. Vokkarane,   Qiong Zhang,   Jason P. Jue,   and   Biao Chen

Department of Computer Science, The University of Texas at Dallas
Richardson, TX 75083
{vinod, qzhang77, jjue, bchen}@utdallas.edu

Abstract– **In this paper, we address the issue of providing differentiated services to IP packets over an optical burst switched core network, and we introduce a new approach for assembling packets into a burst. In this technique, a composite burst is created by combining packets of different classes into the same burst. The packets are placed from the head of the burst to the tail of the burst in order of decreasing class. The performance of this approach is enhanced by using a burst segmentation technique in which, during burst contention, only the packets in the tail of a burst are dropped. We describe a generalized model for burst assembly and burst scheduling, and we propose several composite burst assembly methods. We observe that having multiple class of packets in a burst performs better than having a single class of packets in a burst.**

I. Introduction

The explosive growth of Internet will result in an increased demand for higher transmission rate and faster switching technologies. IP over WDM is a promising framework that can support the bandwidth and flexibility requirement of the next generation networks. In order to efficiently utilize the amount of raw bandwidth in WDM networks, an all-optical transport method, which avoids optical buffering while handling bursty traffic, and which also supports fast resource provisioning and asynchronous transmission of variable sized packets, must be developed. IP over optical burst-switching (OBS) is one such method for transporting traffic directly over a bufferless WDM network [1].

In an OBS network, a data burst consisting of multiple IP packets is switched through the network all-optically. A control header is transmitted ahead of the burst in order to configure the switches along the burst's route. In a delayed reservation (DR) signaling scheme, the burst follows an out-of-band control header after some offset time, without waiting for an acknowledgment for the connection establishment. The offset time allows for the header to be processed at each intermediate 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 [1].

An important issue in OBS networks is contention resolution. Existing contention resolution schemes include deflection, wavelength conversion, and buffering [2]. An approach for reducing packet loss due to contention is burst segmenta-

tion [3], [4]. *Burst segmentation* is the process of dropping only those parts of a burst which overlap with another burst.

Another important issue in OBS networks is burst assembly. Burst assembly is the process of aggregating and assembling input packets into a burst. The most common burst assembly techniques are *timer-based* and *threshold-based*. In timer-based burst assembly approaches, a burst is created and sent into the optical network at periodic time intervals [5]; hence, the network may have variable length input bursts. A threshold is a limit on number of packets contained in each burst before the burst is sent into the optical network. Hence, the network will have fixed-size input bursts.

QoS Support is another important issue in OBS networks. Applications with diverse QoS demands, such as voice-over-IP, video-on-demand, and video conferencing, urge the Internet to guarantee QoS. The IETF has proposed two frameworks in order to support QoS in IP networks: IntServ and DiffServ. IntServ has a per-flow reservation based architecture which is not scalable. In Diffserv, to achieve scalability packets are classified according to the code-point in the IP packet header.

Several solutions have been proposed to support QoS in the OBS core network. In [6], a prioritized offset scheme was proposed to provide QoS in a buffer-less OBS core network. In this offset based reservation scheme, the higher priority bursts are given a larger offset time as compared to the lower priority burst. By providing a higher offset time, the probability of reserving the resources for the higher priority burst is increased and therefore, the loss of higher priority packets is decreased. An alternate approach for providing priority by varying the offset time is to have priority inside the burst header packet (BHP), and to provide differentiated contention resolution based on the burst priority. Burst differentiation policies using burst segmentation are discussed in [4].

In this paper, we focus on the issue of providing QoS through differentiated burst assembly. We introduce a new *composite burst* assembly technique which attempts to meet the delay and loss constraints of the each IP packet class without any extra offset-time. The QoS requirements of an IP packet are defined by the packet's *class*. The QoS requirements are partially fulfilled by assigning *priorities* to bursts, and by providing differentiated contention resolution in the OBS core. We assume that the OBS core supports burst segmentation and a priority-based contention resolution scheme as described in [4]. A DR technique is assumed for signaling and reservation. No fiber
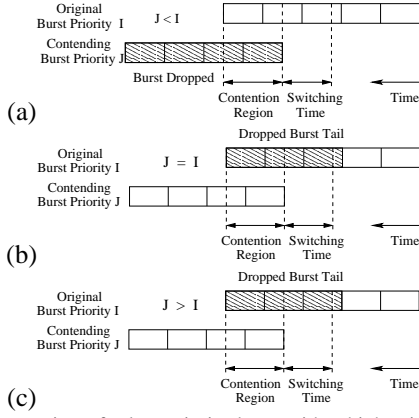
Fig. 1. (a) Contention of a low-priority burst with a high-priority burst. (b) Contention of equal priority bursts. (c) Contention of a high-priority burst with a low-priority burst.

delay lines or wavelength converters are used. We propose a generalized burst assembly framework for mapping different classes of packets into bursts of different priorities. Using this framework, we can characterize a large class of existing burst assembly and scheduling techniques.

The paper is organized as follows. Section II discusses the network architecture of our proposed model, including segmentation policies for QoS support and the current and new techniques for creating bursts. Section III describes the generalized burst assembly and scheduling framework. Section IV describes the proposed burst assembly techniques, and Section V compares the simulation results for different burst assembly schemes. Section VI concludes the paper.

II. NETWORK ARCHITECTURE

A. Contention Resolution Schemes

Contention occurs when two bursts contend for the same output port. In this paper, we will refer to the burst which arrives to the switch first as the *original burst*, and the burst which arrives to the switch later as the *contending burst*. We assume that a prioritized burst segmentation approach [4] is used to handle contentions. In this approach, when a high priority contending burst contends with a low priority original burst, the tail segments of the original burst are preempted and dropped. On the other hand, if a low priority contending burst contends with a high priority original burst, the entire contending burst will be dropped. If two bursts of equal priority contend with each other, then the tail segments of the original burst are dropped. Figures 1(a)-(c) show the three cases of contention involved when burst priorities are supported in the core.

B. Burst Assembly Architecture

To provide QoS support in the optical backbone, the burst assembling and scheduling policies have to take into account the number of classes of packets and the number of burst priorities supported in the core. A burst can contain packets of a particular class (Fig. 2), or a combination of packets of different classes (Fig. 3). Existing burst assembling schemes assemble bursts with packets of the same class. We introduce a new scheme of assembling packets of different classes into a single burst, namely, *composite bursts*.

In practical situations, the number of IP packet classes is greater than the number of burst priorities. Composite burst assembly techniques can provide service differentiation even when the number of packet classes is higher than the number of burst priorities. Also, if the contention resolution policy at the core supports segmentation, it is advantageous to have bursts with multiple packet classes. A burst can be assembled such that the highest-class packets are placed at the head of the burst, while lower-class packets are placed away from the head of the burst in decreasing order of their packet class. This ordering works well with the segmentation-based contention resolution policy, since during a contention, the tail of the original burst, which will consist primarily of lower-class packets, will be dropped. In such a scheme, the lower-class packets are dropped for the benefit of the higher-class packets; thus, the tail of a burst effectively has lower priority than the head of a burst. In the next section, we explain a generalized framework for burst assembly and burst scheduling.

III. GENERALIZED BURST ASSEMBLY AND SCHEDULING FRAMEWORK

In this section we formulate a generalized framework for burst assembly and burst scheduling. The primary issues are which class of packets to put into a burst and when to send the burst into the network.

A. Burst Assembly

Let N be the number of input packet classes and let M be the number of burst priorities supported in the core network. Given N packet classes and M burst priorities, the objective is to meet the QoS requirements by defining a set of *burst types* which specify how packets are aggregated, and by assigning an appropriate burst priority to burst type. In this model, we consider fixed size packets. Also, the length of the burst is the number of packets in a burst. Let K be the number of burst types, where $M \leq K \leq 2^N - 1$. A burst type of type k is characterized by the following parameters:

- L_k^{MIN}: minimum length of burst of type k.
- L_k^{MAX}: maximum length of burst of type k.
- R_{jk}^{MIN}: minimum number of packets of class j in a burst of type k.
- R_{jk}^{MAX}: maximum number of packets of class j in a burst of type k.
- $S_k = \{j \mid R_{jk}^{MAX} > 0\}$: the set of packet classes which may be included in a burst of type k.
- P_k: priority of burst of type k.
- τ_k: timeout value for creating bursts of type k.
- T_k: threshold value for creating bursts of type k.
- C_k: $C_k \subseteq S_k$, subset of packet classes over which the threshold is evaluated. If x_j is the number of packets of class j, then a burst is created if $\sum_{j \in C_k} x_j \geq T_k$.

The burst creation criterion is satisfied, when the threshold value, T_k for each C_k or the timeout value, τ_k for each burst type k is reached. Other class packets in S_k are appended to the burst untill L_k^{MAX} is reached.

For example, if $S_k = \{1, 2\}$, then C_k can be $\{1,2\}$, $\{1\}$ or $\{2\}$. If $C_k = \{1,2\}$, then a burst of type k is created when the sum of the number of packets of class 1 and class 2 is $\geq T_k$. If $C_k = \{1\}$, then a burst of type k is created when the number of packets of class 1 is $\geq T_k$; and the other packet classes that belong to burst type S_k, namely class 2 will be appended to the burst until L_k^{MAX} is reached.

B. Burst Scheduling Techniques

Once a burst is created it must be sent into the OBS core. Burst scheduling is the problem of sending the created bursts into the core such that the loss, delay, and bandwidth constraints of each class are met. In OBS, we must consider the scheduling of electronically buffered burst at the ingress, while simultaneously handling the all-optical transit traffic. Hence, in case of a contention at the source, where the intended output port has been occupied by a transit burst of priority P_x, the burst scheduling policy has to take into account the relative priorities of each new burst versus P_x.

There are many burst scheduling principles. In first-come-first-served (FCFS), bursts are served in the same order that they are created. In priority queuing (PQ), a burst is scheduled to an output port only if all burst queues of the higher priority are empty. In weighted round robin (WRR), each prioritized burst queue is served in a round-robin order. In each round, the number of burst sent is according to the weight assigned by the policy. In waiting time priority (WTP), the priority of a burst is calculated by its waiting time, and the priority increases with waiting time. The scheduler chooses the burst at the head of the queue with the largest priority, and send this burst into the core.

In this paper, the created bursts are sent in FCFS order. In case the outgoing port is occupied by a transit burst, the burst priorities are compared. If the created burst has higher priority than the transit burst, then it preempts the transit burst.

IV. Burst Assembly Techniques

The important design considerations when defining the burst types and scheduling policies are packet loss probability, delay constraints, and bandwidth guarantees. Packet loss probability can be reduced by specifying an appropriate threshold, T_k and burst size L_k^{MAX}. Also, by appropriately mapping packet classes to burst types and by assigning appropriate priority, P_k, to burst types, differenciated levels of packet loss may be achieved. End-to-end delay constraints can be met by setting appropriate timeout values for each burst type, τ_k. Bandwidth guarantees can be provided by choosing an appropriate scheduling policy, and by choosing an appropriate R_{jk}^{MIN} and R_{jk}^{MAX} for each packet class. In this paper, we focus primarily on achieving differentiated loss and delay. A fixed value of T_k, is assigned for all burst types, and a timeout value, T_k is
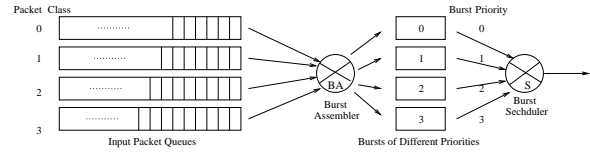

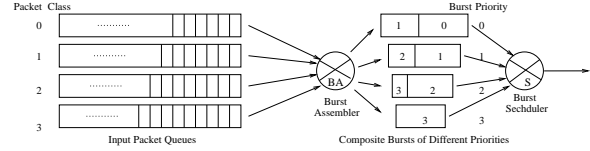Fig. 2. Creation of Single Class Burst with N = 4 and M = 4.

Fig. 3. Creation of Composite Class Burst with N = 4 and M = 4.

assigned only to the highest-priority burst. We investigate the following approaches for selecting mappings S_k and priorities P_k to achieve differentiated QoS.

A. Approach 1: Single Class Burst (SCB) with $N = M$

Let us consider the approach of assembling packets of same class into a burst for the case in which the number of classes is equal to the number of priorities, $N = M$. In order to obtain burst types shown in Fig. 2, we set the following parameters in the generalized framework.

- Set $K = M$
- Set $S_k = \{k\}$, $C_k = S_k$, and $P_k = k$, for $0 \leq k < K$
- Set $L_k^{MIN} = L_k^{MAX} = T_k$, for $0 \leq k < K$
- Set $R_{jk}^{MIN} = R_{jk}^{MAX} = T_k$, for $0 \leq k < K, j = k$
- Set $R_{jk}^{MIN} = R_{jk}^{MAX} = 0$, for $0 \leq k < K, j \neq k$

For example, if $N = 4$ and $M = 4$, we set the number of burst types, $K = 4$. Set $S_0 = \{0\}$, $S_1 = \{1\}$, $S_2 = \{2\}$ and $S_3 = \{3\}$. Here $C_0 = \{0\}$, $C_1 = \{1\}$, $C_2 = \{2\}$ and $C_3 = \{3\}$.

Each packet is buffered in separate queue based on its class. For example, packets of class 2 are collected in an input queue. Once the assembly criterion is satisfied a burst consisting of packets of class 2 is created and sent into the network with a burst priority 2. This process is followed for each class. Thus, the priority of a burst will directly correspond to a specific class of packets contained in the burst.

B. Approach 2: Composite Class Burst (CCB) with $N = M$

In composite bursts, each burst can consist of packets of different classes. In order to obtain burst types shown in Fig. 3, we set the following variables in the generalized framework.

- Set $K = M$
- Set $S_k = \{k, k + 1\}$
- Set $C_k = \{k\}$ and $P_k = k$, for $0 \leq k < K$
- Set $L_k^{MIN} = T_k$ and $L_k^{MAX} = \infty$, for $0 \leq k < K$
- Set $R_{jk}^{MIN} = 0$, for $0 \leq k < K, 0 \leq j < N$
- Set $R_{jk}^{MAX} = \infty$, for $0 \leq k < K, j \in S_k$
- Set $R_{jk}^{MAX} = 0$, for $0 \leq k < K, j \notin S_k$

A burst of type k is generated if the number of packets of class k is equal to the threshold T_k or if the timeout τ_k is reached.
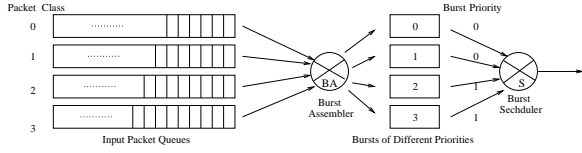
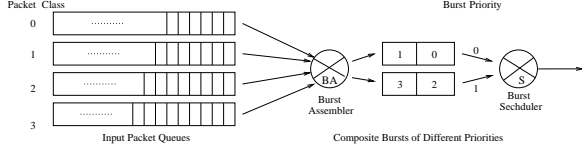Fig. 4. Creation of Single Class Burst with N = 4 and M = 2.

Fig. 5. Creation of Composite Class Burst with N = 4 and M = 2.

This burst contains packets defined by S_k and is generated by appending all constituent class packets into the burst according to the members of S_k in decreasing order of class, such that, the highest class packet in that burst type is at the head of the burst.

For example, if $N = M = 4$, we set the number of burst types, $K = 4$. We select, $S_0 = \{0,1\}, S_1 = \{1,2\}, S_2 = \{2,3\}$, and $S_3 = \{3\}$. Here $C_0 = \{0\}$. $C_1 = \{1\}$, $C_2 = \{2\}$, and $C_3 = \{3\}$. If the threshold of packet class 1 is met, then a burst of type 1 is created with packets of class $S_1 = \{1,2\}$, where class 1 packets are placed at the head of the burst.

C. Approach 3: Single Class Burst (SCB) with $N > M$

We now consider single-class bursts for the case $N > M$. In order to obtain burst types shown in Fig. 4, we have to set the following parameters in the generalized framework.

- Set $K = N$
- Set $S_k = \{k\}$, and $C_k = S_k$, for $0 \leq k < K$
- Set $P_k = \lfloor kM/N \rfloor$, for $0 \leq k < K$
- Set $L_k^{MIN} = L_k^{MAX} = T_k$, for $0 \leq k < K$
- Set $R_{jk}^{MIN} = R_{jk}^{MAX} = T_k$, for $0 \leq k < K, j = k$
- Set $R_{jk}^{MIN} = R_{jk}^{MAX} = 0$, for $0 \leq k < K, j \neq k$

In this case each burst consists of packets of a single class; however, different burst types can have the same burst priority. For example, if $N = 4$ and $M = 2$, we set the number of burst types, $K = 4$. We have four unique types of burst, with each burst assigned one of the two burst priorities. Set $S_0 = \{0\}$, $S_1 = \{1\}$, $S_2 = \{2\}$ and $S_3 = \{3\}$. Here $C_0 = \{0\}$, $C_1 = \{1\}$, $C_2 = \{2\}$ and $C_3 = \{3\}$.

Bursts contains either class 0 or class 1 packets both have priority 0. While bursts containing either class 2 or class 3 packets both have priority 1.

D. Approach 4: Composite Class Burst (CCB) with $N > M$

In order to obtain burst types shown in Fig. 5 we set the following parameters in the generalized framework.

- Set $K = M$
- Set $S_k = \{\frac{kN}{M}, ..., \frac{(k+1)N}{M} - 1\}$, for $0 \leq k < K$
- Set $C_k = S_k$, and $P_k = k$, for $0 \leq k < K$
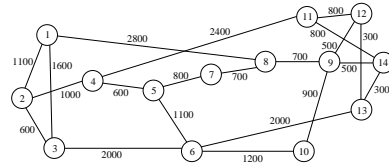- Set $L_k^{MIN} = L_k^{MAX} = T_k$, for $0 \leq k < K$

Fig. 6. NSFNET with 14 nodes.

- Set $R_{jk}^{MIN} = 0$, for $0 \leq k < K, 0 \leq j < N$
- Set $R_{jk}^{MAX} = T_k$, for $0 \leq k < K, j \in S_k$
- Set $R_{jk}^{MAX} = 0$, for $0 \leq k < K, j \notin S_k$

For example, if $N = 4$ and $M = 2$, we set the number of burst types, $K = 2$. We select $S_0 = \{0,1\}$ and $S_1 = \{2,3\}$. Here $C_0 = \{0,1\}$ and $C_1 = \{2,3\}$.

A burst of type k is generated if the sum of packets of classes in C_k is equal to the threshold T_k. Once the T_k or τ_k criterion is met, a burst of type k containing packets defined by S_k is generated by appending all constituent class packets into the burst in the decreasing order of class, such that, the highest-class packet in that burst type is at the head of the burst.

For example, if the threshold of packet class 1 is met, then a burst of type 0 is created with packets of class $S_1 = \{0,1\}$. Hence the two types of composite bursts $\{0,1\}$ and $\{2,3\}$ are assigned burst priorities 0 and 1 respectively.

V. SIMULATION RESULTS

In order to evaluate the performance of the different burst assembly schemes, we develop a simulation model. The following have been assumed to obtain the results: Packet arrivals into the network are Poisson with rate λ. The input traffic ratios of individual packet classes are 40%, 30%, 20%, and 10% for class 3, 2, 1, and 0 respectively. Class 0 is the highest packet class. We set a threshold value of 100 packets for each burst type, and a timeout value of 50 ms for the highest priority burst. The transmission rate is 10 Gbps, and packet length is 1250 bytes. Switching time is 10 μs. The simulations are based on single wavelength in the OBS network. Figure 6 shows the 14-node NSFNET on which the simulation was implemented. The distances shown are in km.

Figure 7(a)-(b) plots packet loss probability and average end-to-end delay versus load for both CCB and SCB with $N = M = 4$. We refer to this case as the 4:4 mapping. We observe that, by using CCB, the loss of packets is more proportional to the packet class than in SCB. We see that the loss of lower-class packets is better in CCB, since some of the lower-class packets are placed into higher priority bursts, which, in turn, decreases the loss probability. Also, the highest-class packets in CCB perform as well as in SCB, since at every contention between highest priority bursts, the lower-class packets are more likely to be dropped. We see that the average delay decreases with the increase in load. This decrease is due to the higher arrival rate of packets which causes the threshold to be satisfied more frequently. The delay of highest-class packets is fairly constant, since we enforce an upper-limit on the aggregation time by using a timeout.
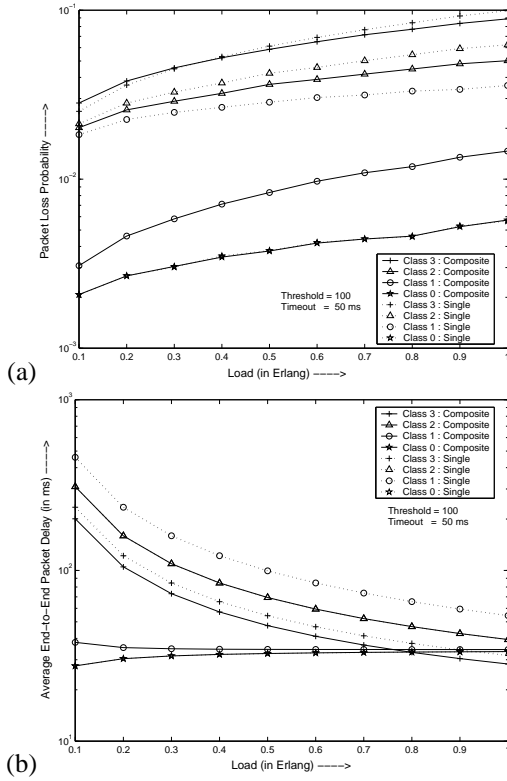
Fig. 7. (a) Packet loss probability versus load and (b) average end-to-end packet delay versus load for N = 4 and M = 4.

Fig. 8. (a) Packet loss probability versus load and (b) average end-to-end packet delay versus load for N = 4 and M = 2.

Figure 8(a)-(b) plots packet loss probability and average end-to-end delay versus load for both CCB and SCB with $N = 4$ and $M = 2$. We refer to this case as the 4:2 mapping. We observe that the performance of CCB is much better than SCB for the highest-class packets. This is due to the fact that in a 4:2 mapping, both packet class 0 and 1 are assigned priority 0, and in an equal priority contention, packets of class 1 may preempt packets of class 0. In SCB, the loss of class 0 packets and class 1 packets will be the same if the input ratio are the same, and if the same threshold and timeout values are used. In our example, a timeout value is assigned to bursts carrying class 0 packets but not to bursts carrying class 1 packets. This difference results in lower loss and delay for class 0 packets, even though the burst are of equal priority. Also, we see that the average end-to-end delay for class 0 and class 1 in the case of CCB, are similar in both 4:4 and 4:2 mapping, since class 1 packets are included in the same bursts as class 0 packets when the timeout is reached. The slight difference in delay between class 0 and class 1 packets is due to their different arrival rates. Also, we observe that the aggregate packet loss of all the packet classes in composite burst assembly is much lower than in single burst assembly. Due to the lack of space we have not included the supporting graph.

VI. CONCLUSION

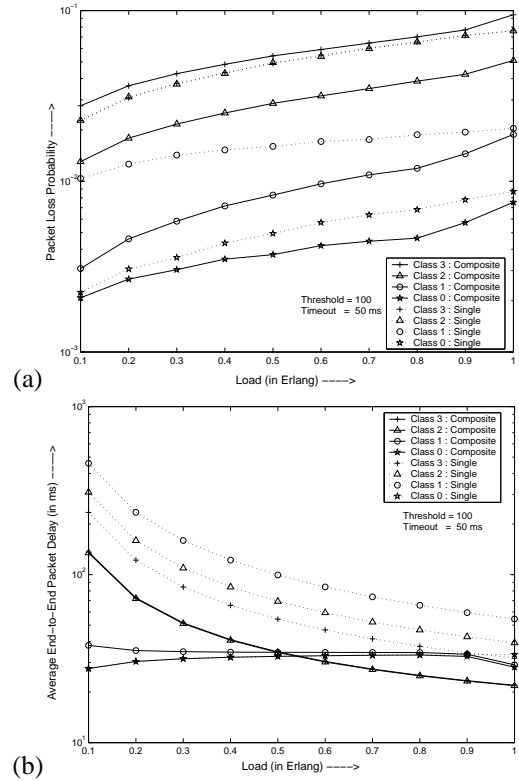In this paper, we consider an OBS network which uses DR signaling scheme with segmentation. We introduced the concept of composite bursts, and we described a generalized framework for burst assembly and burst scheduling. We considered four different burst assembly approaches and evaluated their performance in terms of delay and loss. We observe that approaches with composite bursts perform better than approaches with single-class bursts with respect to providing differentiated QoS for different classes of packets.

An area of future work is to design scheduling policies to satisfy the bandwidth guarantees of each packet class. It is also important to find the optimal burst types and the optimal number of burst priorities in the core for a given number of packet classes with QoS requirements.

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] 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.

[3] V. M. Vokkarane, J. P. Jue, and S. Sitaraman, "Burst Segmentation: An Approach for Reducing Packet Loss in Optical Burst Switched Networks," *Proceedings, IEEE, ICC 2002*, New York, NY, April 2002.

[4] V. M. Vokkarane and J.P. Jue, "Prioritized Routing and Burst Segmentation for QoS in Optical Burst-Switched Networks," *Proceedings, OFC 2002*, Anaheim, CA, March 2002.

[5] A. Ge, F. Callegati, and L.S. Tamil, "On Optical Burst Switching and Self-Similar Traffic," *IEEE Communications Letters*, vol. 4, no. 3 , pp. 98-100, March 2000.

[6] 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.