Reliable Optical Burst Switching for Next-Generation Grid Networks

Vinod M. Vokkarane [*] and Qiong Zhang [+]

[*] Department of Computer and Information Science, University of Massachusetts at Dartmouth, North Dartmouth, MA 02747, USA

[+] Department of Mathematical Sciences and Applied Computing, Arizona State University (West Campus), Glendale, AZ 85306, USA

E-mail: vvokkarane@umassd.edu and qiong.zhang@asu.edu

***Abstract*— Grid computing provides a flexible, secure, sharing of resources among dynamic collections of individuals, institutions, and resources. Optical burst switching (OBS) is one of the most promising new optical transport paradigms for efficiently transporting data over an all-optical network. In this paper, we discuss several mechanisms for deploying a reliable OBS network using a range of loss minimization and loss recovery mechanisms. We also propose two new loss recovery mechanisms for deploying a reliable OBS network, namely, forward error correction and composite burst assembly. Our simulation shows that the two proposed mechanisms significantly reduce the loss experienced by loss-sensitive connection-oriented traffic through an OBS network. We have also shown how the reduction of loss leads to increased throughput of TCP-over-OBS network. We also propose a comprehensive OBS service architecture, and provide guidelines for implementing a reliable connection-oriented photonic transmission control protocol (PTCP) over OBS-based Grid networks.**

***Keywords:* Grid Computing, WDM, TCP, IP, OBS, and FEC.**

I. Introduction

Grid computing provides a flexible, secure, sharing of resources among dynamic collections of individuals, institutions, and resources [1]. From an application perspective, there are two types of grids: computational grids and data grids. A compute grid is essentially a collection of distributed computing resources, within or across locations that are aggregated to act as a unified processing resource or virtual supercomputer. These computational resources can be either within or between administrative domains. The benefit is faster, more efficient processing of computationally-intensive jobs, while utilizing existing resources. Computational grids also eliminate the drawback of tightly binding specific machines to specific jobs, by allowing the aggregated pool to most efficiently service sequential or parallel jobs with fine-grained user attributes. A data grid on the other hand provides wide area, secure access to current data. Data grids enable users and applications to manage and efficiently use database information from distributed locations. Data grids can be deployed within one administrative domain or across multiple domains. Data grids eliminate the need to unnecessarily move, replicate, or centralize data, translating into cost savings. Initial data grids are being constructed today, primarily serving collaborative research communities. Software vendors and large enterprises are currently investigating data grid solutions and services for business applications. Down the road, data grids will be a key element in the rollout of Web services. From a networking perspective, the impact of data grids will include a tighter integration of storage protocols and high-performance networking.

Grid computing can be applied to many areas, such as [2]:

- Data intensive core services: enable secure access to massive amounts of data in a global name space, to move and to replicate data at high-speed from one geographical site to another.
- eHealth remote screening: secure access and fast transfer of large screening images on demand within an acceptable latency.
- Bio-applications: provide an integrated environment on a grid for the phytogenetic analysis, the comparative analysis at a large scale, and the synthesis and pattern matching of proteins in the area of bioinformatics.

Optical networks are a logical choice for supporting grid networks in order to ensure global reach and huge amount of inexpensive bandwidth for large file transfer demands, since optical fiber links offering huge bandwidths on the order of 50 THz. In optical wavelength division multiplexed (WDM) networks, channels are created by dividing the bandwidth into a number of wavelength or frequency bands, each of which can be accessed by the end-user at peak electronic rates. In order to efficiently utilize this bandwidth, we have to design efficient transport architectures and protocols based on the state-of-the-art optical device technology [3]. The general requirements that the underlying optical network has to support in order to host grid services are the following:

- Scalable, flexible, and reconfigurable network infrastructure.
- Ability to support very high capacity: Bulk data transfer.
- Bandwidth-on-demand capabilities for short or long periods of time between different discrete points

0-7803-9277-9/05/$20.00/©2005 IEEE

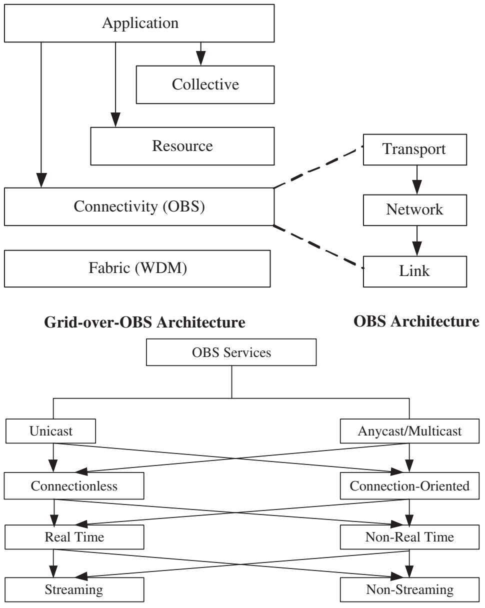
across the network.
- Variable-rate and constant-rate bandwidth services.
- Wavelength and sub-wavelength level resource provisioning.
- Broadcasting/multicasting/anycasting capabilities: so as to assign grid job requests on to multiple grid resources.
- High resilience across layers: for instance, a resilient physical layer will entail a number of features including resilient wavelengths, fast and dependable restoration mechanisms, as well as routing diversity stipulations being available to the grid user.

Current grid deployments support only applications that require long-lived wavelength paths between the client and grid resources. These long-lived wavelength paths are provided by dedicated wavelengths using optical circuit switching (OCS) [4], [5], [6]. In general, a pure OCS system is not bandwidth efficient since majority of the traffic flows do not transfer a fixed continuous amount of data over a long periods (minutes to months). Also, with the increase in channel capacity from OC-3 to OC-768 and above, there is increased pressure to implement photonic transport protocols that support statistical multiplexing of fiber links. Optical Burst Switching (OBS) [7], [8] is a promising switching technology that efficiently utilizes the optical fiber bandwidth provided by wavelength division multiplexing, and at the same time, avoids the need for optical buffering while handling bursty traffic. In a OBS network, a data burst consisting of multiple data packets is switched through the network all-optically. A Burst Header Packet (BHP) is transmitted ahead of the burst in order to reserve the data channel and configure the switches along the burst's route. In the Just-Enough-Time (JET) signaling scheme [8], the burst transmission follows an out-of-band BHP after a predetermined offset time. The offset time allows the BHP to be processed before the burst arrives at the intermediate nodes; thus, the burst does not need to be delayed at the intermediate nodes. The BHP also specifies the duration of the burst in order to let a node know when it may reconfigure its switch for the next arriving burst. Other OBS signaling techniques, such as Just-In-Time (JIT) [9] are also implemented in a one-way unacknowledged manner.

OBS has the potential of meeting several important objectives of Grid services:
- High bandwidth, low latency, deterministic transport required for high demand Grid applications;
- All-optical data transmission with ultra-fast user/application-initiated light path setup;
- Implementable with cost effective COTS (commercial off-the-shelf) optical devices.

Several works have proposed a Grid-over-OBS infrastructure in [10], [11], [12]. Fig. 1 shows the Grid-over-OBS architecture, in which there are two interfaces to connect Grid and OBS network, namely, Grid User Network Interface (GUNI) and Grid Resource Network Interface (GUNI). One critical issue in OBS networks for supporting grid services is that OBS currently provides only unreliable data transfer due to the one-way based nature of all the signaling and reservation techniques. The existing solution is to rely on the higher layer, such as TCP, to provide reliable data transfer, while the OBS layer remains unreliable.

Figure 2(a) shows the protocol layers in a Grid-over-OBS network. The responsibilities of each Grid layer are listed as follows:
- Fabric: provides the resources to which shared access is mediated by Grid protocols.
- Connectivity: defines core communication protocols required for Grid-specific network transactions.
- Resource: define protocols (and APIs and SDKs) for the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources.
- Collective: contains protocols and services that are not associated with any one specific resource but rather are global in nature and capture interactions across collections of resources.
- Application: comprises the user applications that operate within a Virtual Organization environment.



Grid-over-OBS Architecture **OBS Architecture**

Supported Services in OBS Networks for Grid Computing

Fig. 2. (a) Grid-over-OBS Protocol Layering. (b) OBS-supported services.

There is a tremendous need to support reliable connection-oriented end-to-end transport service for supporting application such as the grid systems. TCP has been widely used as the reliable data transport protocol of

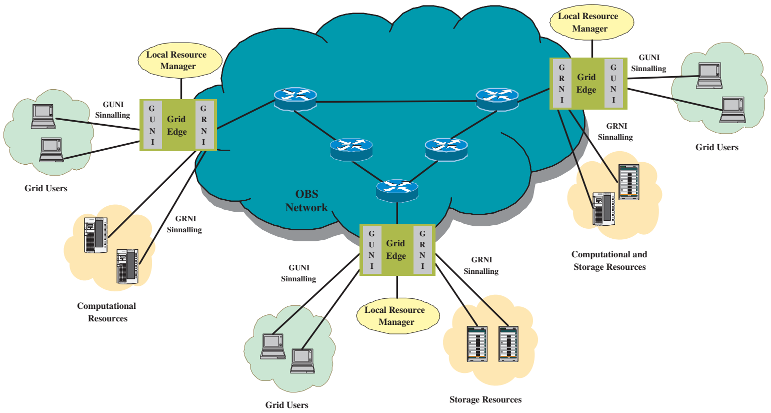


Fig. 1. Grid-over-OBS Architecture

choice for the traditionally electronic Internet. Over the years, TCP has undergone significant changes in terms of developing new congestion-control techniques and handling issues concerning the need for high-bandwidth at the presence of long end-to-end delays between the senders and the receivers. The important TCP flavors currently in existence are TCP Tahoe, TCP Reno, TCP New Reno, TCP SACK, TCP Vegas, High-Speed TCP, Fast TCP, and XCP [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]. The fundamental assumption of all these TCP techniques is that the underlying medium is electronic in nature, and that the packets experience queueing (buffering) delays during congestion in the electronic IP routers along the path of the TCP flow.

In this paper, we develop a new architectural and protocol framework for the next-generation Grid networks, wherein grid-based edge nodes directly connect to the all-optical core network. In such a framework, since there is no know methodology for storing optical signals, once the data is sent into the core, the data cannot be buffered along any of the intermediate nodes in the core network and hence does not experience any queueing delay. Hence, the fundamental data transmission principles of TCP are not applicable to an all-optical transport network. We now need investigate new photonic transport protocols for carrying high-bandwidth grid traffic over an all-optical burst-switched core network.

The data transmission rate of the TCP segments into the core network is represented in terms of the sending rate (or the throughput) of the flow. In the optical network framework there is insignificant variations in the propagation delay between the sender and the receiver of a transport level (TCP) flow, due to the absence of electronic buffers in the core. In this scenario of supporting transport flows with a strictly bounded delay limit, many of the fundamental TCP congestion-control design principles will be invalid. Also, TCP congestion-control techniques that are delay-based, such as TCP Vegas, and TCP flavors that use certain round trip delay estimation may not perform effectively.

We believe that a photonic transport layer that clearly understands the characteristic of the lower optical layer is essential to achieve high data rates. We believe that a comprehensive set of transport services listed in Fig. 2(b) should be supported by the photonic transport layer in order to handle the plethora of different Grid applications. Similar to TCP, there is a need for designing and developing a reliable connection-oriented photonic transmission control protocol (PTCP) over OBS networks. In this paper, we focus on making the OBS (link and network) layer reliable and do not get into the details of the implementation of the OBS transport layer. We believe a simple selective-repeat based sliding-window protocol [23] can be used to support reliable in-order connection-oriented service, while the traditional one-way OBS signaling is suitable for supporting connectionless service. The detailed design and evaluation of photonic transport protocols is out-of-scope of this paper.

II. Transmission Control Protocol (TCP): Background and Issues

There are three different types of TCP congestion control techniques, they are loss-based, delay-based, and rate-based. The well-deployed TCP flavors, Reno [24], [25], New Reno [26], and SACK [27], are loss-based TCP, which uses packet losses to estimate the available bandwidth in networks. However, implementing those loss-based TCP flavors over OBS networks may cause *False Time Out* (FTO), where the network congestion detected by TCP (at low loads) is actually caused by random burst contention instead of IP router buffer overflow. FTO has been shown to significantly degrade the TCP performance. The paper [28] has proposed a new Burst TCP (BTCP) that can detect FTOs and accurately react network congestion.

The delay-based TCP flavors, such as TCP Vegas [18], use delay measurement to estimate available bandwidth in networks. The queueing delay measured in TCP can provide multi-bit information about the degree of network congestion, which will make TCP implementation easier

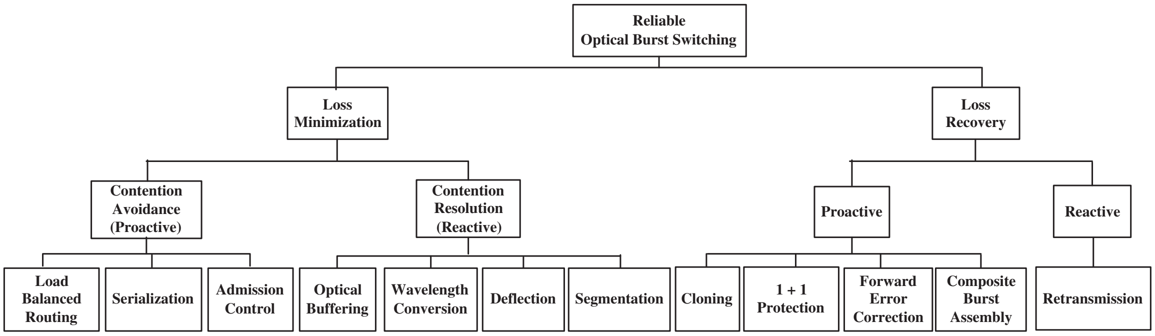


Fig. 3. Reliable OBS Framework.

to stabilize a network with a target fairness and high utilization. The performance of TCP Vegas has been evaluated in [18], [29]. TCP Vegas improves TCP efficiency by achieving 37% to 71% higher throughput and by reducing much less packet retransmissions than TCP Reno.

As bandwidth-delay product (BDP) increases in high-speed networks, several issues arise:

- Inefficiency: linear increase of congestion window size of one per-round-trip-time (RTT) limits the ability to acquire spare bandwidth; and multiplicative decreasing congestion window size per loss event is too drastic.
- Oscillatory: oscillation becomes unavoidable in high-capacity or large-delay networks because TCP uses a binary congestion signal (pack loss).

The basic TCP flavors cannot meet the requirement of Grid service due to the BDP problem. Recently, some new TCP flavors have been proposed in order to solve the BDP problem. HighSpeed TCP is a loss-based TCP which allows TCP to achieve high throughput with more realistic packet drop rate requirement. Fast TCP can be thought of as a high-speed version of TCP Vegas that uses both packet delay and packet loss as a measurement of network congestion.

Recently, a third kind of TCP congestion control mechanisms that are rate-based have been proposed. A rate-based eXplicit Control Protocol (XCP) has been proposed in [14], where available network bandwidth is estimated based on the explicit feedbacks from routers in the networks.

OBS primarily implements one-way, unacknowledged signaling, such as JET. The data bursts in the bufferless OBS core network are susceptible to random burst loss due to burst contentions. This type of random loss is unique to OBS, since in traditional electronic IP networks loss is primarily due to buffer overflow at the core IP routers. Traditionally, TCP implements congestion control and flow control mechanisms to avoid network congestion and to provide in-order reliable data transfer. We believe that a reliable photonic transport protocol does not need to implement congestion control mechanisms but only flow control and rate control mechanisms at the edge. Since, no congestion can occur in an all-optical bufferless core network, unlike electronic (buffer-based) networks. Note that though the network throughput does not collapse with increasing load as in IP networks, the increase in load leads to increase in data loss in all-optical networks. Congestion can only occur at ingress and egress node buffers. Hence, for reliable data transfer in OBS we need flow control mechanisms rather than congestion control mechanisms. A simple sliding window mechanism will suffice in order to ensure reliable in-order delivery. In the future, we intend to evaluate all these three kinds of data transmission mechanisms over a bufferless OBS network.

III. Reliable OBS

In this paper, we focus on the goal of implementing a reliable optical burst-switched network using primarily loss minimization and loss recovery mechanisms. One of the primary OBS core network issues is contention resolution. When two or more bursts are destined for the same output port at the same time, contention occurs. When a contention cannot be resolved, one of the contenting burst is lost. In order to handle burst loss due to unresolved contentions, we implement loss recovery mechanisms. In the following section, we classify and describe the different loss minimization and loss recovery mechanisms. The entire framework for supporting a reliable OBS network is shown in Fig. 3.

A. Loss Minimization: Contention Resolution Vs. Contention Avoidance

We classify all loss minimization mechanisms into two broad categories, namely, *Contention Resolution* and *Contention Avoidance*. Contention resolution mechanisms attempt to minimize data loss when a contention has already occurred. On the other hand, contention avoidance mechanisms attempt to minimize the occurrence of contentions. The contention resolution mechanisms are optical buffering, wavelength conversion, deflection routing, and segmentation. While the contention avoidance mechanisms are load-balanced routing, serialization (proactive scheduling), and admission control. A combination of loss minimization mechanisms may be used to further reduce the data loss.

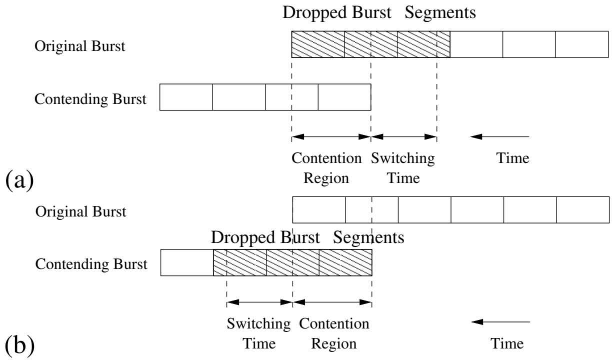


Fig. 4. Selective segment dropping for two contending bursts (a) tail-dropping policy (b) head-dropping policy.

1) Contention Resolution Mechanisms: The primary contention resolution mechanisms are optical buffering, wavelength conversion, deflection routing, and burst segmentation [30], [31]. In optical buffering, fiber delay lines (FDLs) are used to delay the burst for a specified amount of time, proportional to the length of the delay line, in order to avoid the contention [32]. In wavelength conversion, if two bursts on the same wavelength are destined to go out of the same port at the same time, then one burst can be shifted to a different wavelength [33]. In deflection routing, one of the two bursts will be routed to the correct output port (primary) and the other to any available alternate output port (secondary). The deflected packets may end up following a longer path to the destination, leading to higher end-to-end delay, and packets may also arrive at the destination out-of-order [34], [35]. In burst segmentation [31], the burst is divided into basic transport units called *segments*. Each of these segments may consist of a single IP packet or multiple IP packets, with each segment defining the possible partitioning points of a burst when the burst experiences contention in the optical network. All segments in a burst are initially transmitted as a single burst unit. However, when contention occurs, only the overlapping segments of a one of the bursts in contention will be dropped, as shown in Fig. 4. If switching time is not negligible, then additional segments may be lost when the output port is switched from one burst to another. There are primarily two approaches for dropping burst segments during a contention. The first approach, tail dropping, is to drop the tail of the original burst (Fig. 4(a)), and the second approach, head dropping, is to drop the head of the contending burst (Fig. 4(b)) [31].

2) Contention Avoidance Mechanisms: In optical burst-switched networks, there have been several solutions to resolve contentions in order to minimize data loss as we discussed above. These localized contention resolution techniques react to contentions, but do not address the more fundamental problem unbalanced loading of different core links. In [36], two dynamic load-balanced routing techniques are proposed to avoid burst contentions. The simulation results show that the proposed contention avoidance techniques improve the network utilization and reduce data loss. In [37], [38], [39], the authors investigated similar load-balancing routing (or path switching) approaches using adaptive alternate path routing and concluded with similar observations as [36].

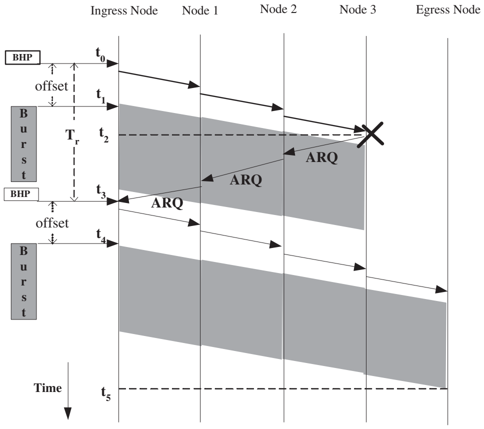


Fig. 5. OBS retransmission scheme.

In [40], a proactive scheduling algorithm referred to as burst overlap reduction algorithm (BORA) is proposed. The motivation behind BORA is based on the observation that if the total number of simultaneously arriving bursts at an output port exceeds the number of channels at that port, burst loss will be inevitable. Thus, if we can reduce the total number of simultaneously arriving bursts from a given source at each port, it is likely that the burst loss will be reduced. BORA tries to proactively avoid burst contention at remote (downstream) nodes. The basic idea is to serialize the bursts on outgoing links to reduce the burst overlapping degree (and thus burst contention and burst loss at downstream nodes). This can be accomplished by judiciously delaying locally assembled bursts beyond the pre-determined offset time using electronic memory available at ingress nodes. Results show that loss rate of BORA is much lower than loss in existing algorithms. The biggest side-effect of BORA is that it introduces significant delay at the edge during serialization of bursts.

In addition, several other edge-based admission control techniques can be incorporated to minimize the number of contentions in the core.

B. Loss Recovery: Reactive Vs. Proactive

Loss recovery mechanism are essential to support reliable data transfer in a OBS network. We classify all loss recovery mechanisms into one of two categories, namely, *Reactive* and *Proactive*. Reactive loss recovery mechanisms are generally optimistic about the successful reception of the transmitted burst at the destination. Hence, reactive mechanisms only attempt to recover when they receive an explicit failure message. On the other hand, proactive loss recovery mechanisms are generally pessimistic about the successful reception of the transmitted burst at the destination. Proactive mechanisms transmit additional information (overhead) along with the original burst so as to handle certain loss scenarios. Broadly speaking, reactive mechanisms are better solutions when burst losses are rare and bandwidth needs to be optimized. Proactive mechanisms are better solutions when burst losses are high and delay needs to be optimized.

We now describe the different loss recovery mechanisms shown in Fig. 3. We first describe the *retransmission* scheme, the only reactive OBS loss recovery mechanisms. We then describe several proactive loss recovery mechanisms, such as burst cloning, $1 + 1$ protection, forward error correction, and composite burst assembly. Note that a combination of loss recovery mechanisms can be implemented to further reduce the loss in the OBS network.

1) Retransmission: The basic idea of burst retransmission is to allow contending bursts to be retransmitted in the OBS layer. In this scheme, BHPs are sent out prior to data burst transmission in order to reserve resources. After an offset time, the burst is transmitted. At the same time, the ingress node stores a copy of the transmitted burst for possible retransmissions. As the BHP traverses through the core nodes, if the channel reservation fails due to burst contention, the core node will send an *Automatic Retransmission Request* (ARQ) to the ingress node in order to report the reservation failure. Upon receiving an ARQ, the ingress node retransmits the corresponding duplicate preceded by its BHP.

We illustrate a retransmission scenario in Fig. 5. In this figure, the BHP is transmitted at time t_0, while the burst is duplicated and stored at the ingress node before being transmitted. The burst is transmitted at time t_1 after some offset time. At t_2, the burst reservation fails at Node 3, triggering Node 3 to send an ARQ back to the ingress node. The ingress node receives the ARQ at t_3, then sends a new BHP and retransmits a duplicate burst at t_4 after some offset time. Assuming the second transmission is successful, at t_5 the burst arrives at the egress node. A burst duplicate may be retransmitted multiple times until the burst successfully reaches the egress node.

We observe from Fig. 5 that the retransmission scheme results in an extra delay, T_r, referred to as *retransmission delay*. The retransmission delay is the time elapsed between the initial BHP transmission of a burst and the last ARQ receipt for the corresponding burst, i.e., $t_3 - t_0$. The retransmission delay can be bounded by a delay constraint, notated as δ. Once the ingress node receives an ARQ for the contending burst, the ingress node calculates T_r for the contending burst and decides if it is necessary to retransmit the burst. If $T_r \geq \delta$, the ingress node ignores the ARQ and does not retransmit the contending burst.

If the network is lightly loaded, the retransmission scheme has a good chance of successfully retransmitting contending bursts. If the network is heavily loaded, the retransmitted bursts have a lower probability of being successfully received. The ingress node can continue to attempt retransmission until the retransmission delay exceeds the delay constraint, in which case the burst is dropped and no longer retransmitted when a contention occurs. Compared to a OBS network without burst retransmission, a OBS network with burst retransmissions will have a higher traffic load in the network, leading to higher burst contention probability. However, the burst is allowed to experience multiple contentions, which leads to a lower burst loss probability, particularly at lower loads. Additional details about retransmission can be found in [41]. We now discuss the proactive loss recovery mechanism (refer Fig. 3).

2) Burst Cloning: In burst cloning [42], the authors propose a proactive loss recovery scheme for OBS networks. The idea is to replicate a burst and send duplicated copies of the burst through the network simultaneously. If any one of the bursts is lost, the destination egress nodes can recover from the core loss using the other duplicate burst. Note that we need some additional intelligence in the BHPs to identify duplicates in the case both original and duplicate burst reach the destination. So that the destination will select one of the bursts, disassemble the burst, and forward the packets on to the corresponding destination hosts. Based on the load on the different link in the network, the original and the clone could be sent on different paths. Primary design issues in burst cloning are to select the optimal node at which to clone and to prevent cloned bursts from contending for resources with their original bursts.

3) $1 + 1$ Protection: $1 + 1$ protection for OBS is discussed in [43]. Here, premium data traffic is protected by routing two copies of the data over disjoint paths. The authors show that a sufficiently large difference in the propagation delays can cause performance degradations that may result in an unsatisfactory quality-of-service on the protected connection. It is important to note that source burst cloning is very similar to $1 + 1$ protection where in cloning performed at the ingress node and the burst are transmitted along disjoint paths to the destination. In this paper, we do not evaluate either burst cloning or $1+1$ protection. Interested readers are referred to [42] and [43].

4) Forward Error Correction (FEC): Forward Error Correction (FEC) is a type of error correction which improves on simple error detection schemes by enabling the receiver to correct errors once they are detected. Furthermore, FEC codes can ameliorate or even eliminate the need for feedback from receivers to senders to request retransmission of lost packets. FEC works by adding check bits to the outgoing data stream. Adding more check bits reduces the amount of available bandwidth, but also enables the receiver to correct for more errors. FEC makes it

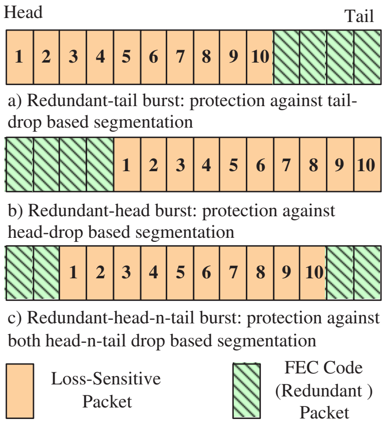

Head
Tail
1
2
3
4
5
6
7
8
9
10
a) Redundant-tail burst: protection against tail-drop based segmentation
b) Redundant-head burst: protection against head-drop based segmentation
c) Redundant-head-n-tail burst: protection against both head-n-tail drop based segmentation
Loss-Sensitive Packet
FEC Code (Redundant) Packet

Fig. 6. Different burst assembly options using FEC, based on the segmentation technique incorporated in the OBS core.

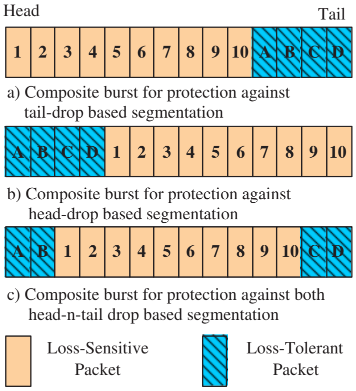

Head
Tail
1
2
3
4
5
6
7
8
9
10
A
B
C
D
a) Composite burst for protection against tail-drop based segmentation
b) Composite burst for protection against head-drop based segmentation
c) Composite burst for protection against both head-n-tail drop based segmentation
Loss-Sensitive Packet
Loss-Tolerant Packet

Fig. 7. Different burst assembly options using CBA, based on the segmentation technique incorporated in the OBS core.

possible to transmit at much higher data rates if additional bandwidth is available. FEC is particulary well suited for optical transmissions, where bandwidth is reasonable but end-to-end latency across long-haul networks is significant.

In a communication system that employs forward error-correction coding, a digital information source sends a data sequence to an encoder. The encoder inserts redundant (or parity) bits, thereby outputting a longer sequence of code bits, called a *codeword*. Such codewords can then be transmitted to a receiver, which uses a suitable decoder to extract the original data sequence. FEC codes can be implemented using several different approaches, such as block codes and convolutional codes. In block coding, the encoder intersperses parity bits into the data sequence using a particular algebraic algorithm. On the receiving end, the decoder applies an inverse of the algebraic algorithm to identify and correct any errors. While convolutional codes process the incoming bits in streams rather than in blocks. The paramount feature of such codes is that the encoding of any bit is strongly influenced by the bits that preceded it.

Recently, [44] proposed Reed-Solomon (RS) code based FEC mechanism to provide protection in a OBS network wherein bits in multiple bursts are encoded to create a redundant burst (of RS codes), and all these bursts are transmitted on multiple paths in order to provide protection against links failures in the core network. RS codes are represented by (n, k), where n byte data consists of k byte original data and $(n-k)$ byte redundant code. The (n, k) RS code can recover up to $(n-k)$ byte loss [45]. The FEC burst loss recovery scheme in [44] has limited scope since it is only applicable to traffic that need transmission of multiple bursts. Also, in order to implement this FEC mechanism, all these group of bursts have to sent on several (possibly disjoint) paths across the network. Transmitting burst on multiple paths causes severe problems at the receiver in terms of buffering delay so as to reorder, decode, and verify these bursts.

In this paper, we propose a FEC-based loss recovery technique for a OBS network with burst segmentation support. As discussed before, segmentation drops only the overlapping packets of a burst in contention to minimize packet loss. In our scheme, FEC codes (or redundant packets) can be placed along with every burst so that the receiver can recover from packet loss of each burst in the forward direction. Note that without segmentation, there is no benefit of adding redundant FEC codes into a burst. Based on the type of segmentation [31] implemented by the OBS core, we can place the correction packets at specific positions inside a burst as shown Fig. 6.

Figure 6 illustrates the specific locations of a burst that are more susceptible to loss for each flavor of burst segmentation implemented in the core. For instance, if in a (n, k) RS code with $n = 14$ and $k = 10$, we could place the four redundant FEC code packets toward the tail of the burst, as shown in Fig. 6(a), if a strict tail-dropping segmentation is implemented in the OBS core. Similarly, Fig. 6(b) and Fig. 6(b) depict the scenario if a strict head-dropping and a head-n-tail-dropping is implemented in the OBS core. In the general case, segmentation can be performed on the tail, the head, or both, and no specific burst assembly mechanism would do better than the other. We could to implement a RS-code based FEC technique to recover from errors.

We observe that if the core follow a specific type of segmentation, there is no need to waste the bandwidth with redundant FEC code packets. Instead, we could assembly burst consisting of packets from multiple traffic streams, so that we place loss-tolerant packets at positions that are more susceptible to packet loss compared to loss-sensitive packets. We now explain a composite burst assembly technique for minimizing packet loss in a segmentation-based OBS network.

5) Composite Burst Assembly (CBA): Composite burst assembly combined with burst segmentation is an effective mechanism for loss recovery in OBS networks. We observe that, if we are knowledgeable about the manner in which the packets within a burst gets dropped due to segmentation, we can implement a better bandwidth-efficient loss recovery mechanism called *Composite Burst Assembly (CBA)* [46]. In FEC, we add overhead data to correct from errors, while in CBA we do not add any overhead data but assembly the original burst in a composite manner such that it reduces loss probability of certain class of packets.

Figure 7 illustrates the motivation for forward error prevention mechanisms in segmentation-based OBS networks. We consider three segmentation scenarios that a burst can experience in the core network as discussed before. For a detailed discussion of how to implement a OBS network with a strict tail-dropping, strict head-dropping, or both head-n-tail dropping (non-preemptive) policy refer to [31], [47]. In CBA, the ingress nodes performs composite burst assembly wherein, different traffic flows are assembled together such that the packets belonging to loss-sensitive traffic (e.g. connection-oriented) are placed at positions of low loss probability, while packets belonging to loss-tolerant traffic (e.g. connectionless) are placed at positions of higher loss probability. In Fig. 7, the shaded regions (in blue) represent the positions of high loss probability, while the solid regions (in brown) represent the positions of low loss probability for each type of segmentation. We envision connection-oriented traffic, to be assembled with connectionless traffic, and get similar (if not better) loss performance as compared to FEC for connection-oriented traffic. Hence, given the packet loss pattern inside a burst, CBA will outperform FEC in terms of bandwidth-efficiency with no additional delay.

The only tradeoff that may result from creation of composite bursts is that the loss experienced by connectionless traffic may be higher at significant loads. Without retransmission support, certain high-bandwidth real-time connectionless traffic (for instance, high-quality video streaming) may need to be protected from high loss. In such cases, those specific connectionless traffic may need to be assembled into separate bursts.

IV. Simulation Results

In this section, we develop a network-wide simulation model in order to evaluate the performance of loss minimization and loss recovery mechanisms. We evaluate the performance of FEC, CBA, Retransmission, Segmentation, and compare them to a baseline scheme that drops the entire burst when burst contentions occur. We simulate the NSF network as shown in Fig. 8. The number of wavelengths on each link is 8 and the transmission rate on a wavelength is 10 Gb/s. We assume that all core nodes are bufferless (no FDLs) and have full-wavelength conversion capability. The data traffic simulated traverses through eight ingress-egress node pairs: (1,11), (3,11), (2,9), (3,9), (1,13), (2,10), (4,12), and (7,13). Burst arrivals follow a Poisson process and are uniformly distributed among the eight flows. Each burst generated has a fixed length of 100 packets and each packet is 1250 bytes long. The load in each figure is the original input traffic load to the entire network in Erlang.

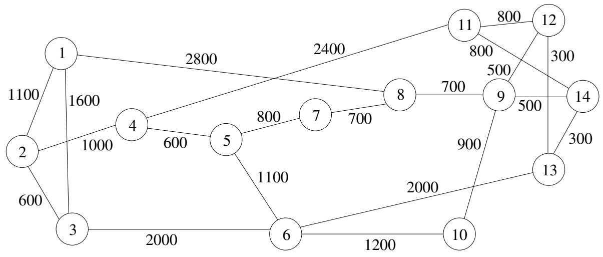


Fig. 8. NSF network.

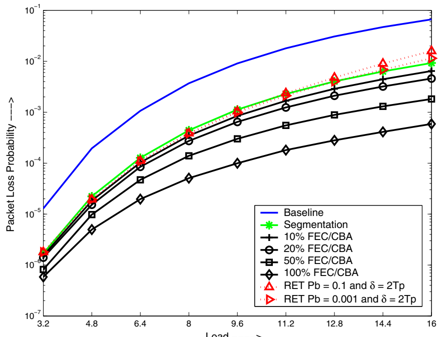


Fig. 9. Packet loss probability of loss-sensitive traffic vs. load.

Figure 9 plots the average packet loss probability of loss-sensitive traffic versus load for the OBS network with different loss minimization and loss recovery mechanisms. In the burst retransmission scheme, we set the delay constraint to be $2T_p$ and the different retransmission buffer blocking probabilities to be 0.1 and 0.001. For the loss-sensitive traffic, packet loss probabilities in the FEC and CBA schemes are the same. We can see that, with higher redundancy, the packet loss probability of FEC/CBA schemes reduces. We can also see that the FEC/CBA schemes performs better than the segmentation, burst retransmission, and baseline schemes. Note that $x\%$ FEC refers to burst created with $x\%$ redundant FEC code packets and $x\%$ CBA refers to composite burst with $x\%$ of loss-tolerant traffic packets.

Figure 10 plots the average packet loss probability of loss-tolerant traffic versus load for the OBS network with the CBA scheme. We compare the CBA scheme with different redundancy values. We can see that, in a specific load range, there exists a redundancy value using which the CBA scheme performs the best. For example, at the loads below 9.6 Erlang, the CBA scheme with 20% redundancy has the lowest packet loss probability; at the loads between 11.2 and 14.4 Erlang, the CBA scheme with 30% redundancy has the lowest packet loss proba-

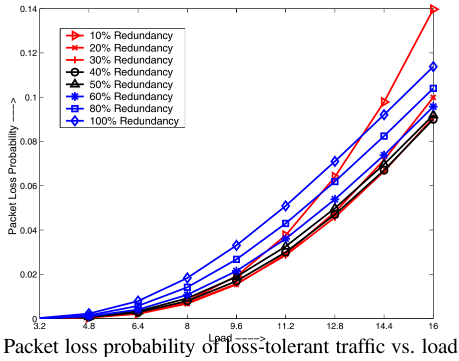


Fig. 10. Packet loss probability of loss-tolerant traffic vs. load.

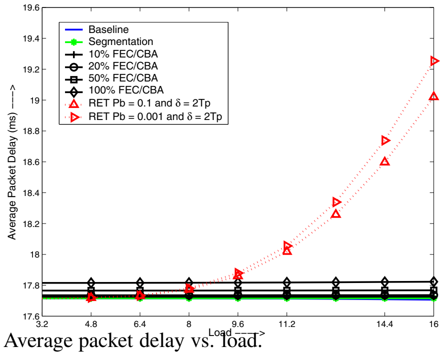


Fig. 11. Average packet delay vs. load.

bility; and at the load of 16, the CBA scheme with 40% redundancy performs the best. Hence, we can design a dynamic CBA scheme that adjusts the redundancy based on traffic load range in order to achieve optimal loss for both loss-sensitive and loss-tolerant traffic.

Figure 11 plots the average packet delay versus load for the OBS network with the different loss minimization and loss recovery mechanisms. We can see that the retransmission scheme has the highest average packet delay due to the retransmission delay. The delay incurred in the FEC/CBA schemes only includes one-way propagation delay and data transmission delay. We see that the packet delay in the FEC/CBA scheme is only a little higher than the baseline scheme. Also, FEC/CBA with higher redundancy results in higher packet delay, since higher redundancy generates larger-sized bursts resulting in higher data transmission delay.

Figure 12 plots the send rate for TCP Reno over the OBS network using the FEC/CBA scheme. The TCP Reno send rate is calculated based on the throughput equation modeled in [48]. We can see that, using the FEC/CBA scheme, the TCP throughput increases significantly compared to segmentation and baseline schemes due to improved packet loss probability. We can also see that, the TCP throughput increases when using higher redundancy in the FEC scheme, since higher redundancy improves packet loss probability in the OBS network.

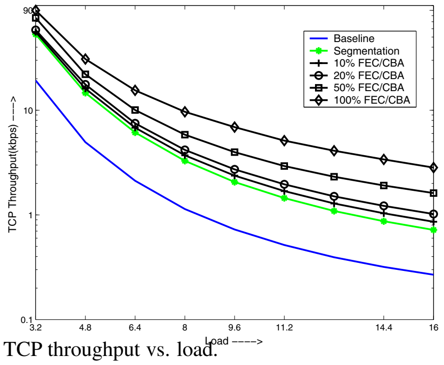


Fig. 12. TCP throughput vs. load.

V. Conclusion

In this paper, we present the concept of implementing next-generation grids over a reliable OBS network. We described the Grid-over-OBS network architecture along with the different protocol and service layers. Also, a comprehensive framework for implementing a reliable OBS network using loss minimization and loss recovery mechanisms was presented. The different proactive and reactive mechanisms are introduced and evaluated. We also compare the performance of the FEC and CBA schemes with burst retransmission scheme using the NSF network. Our simulation results show that both FEC and CBA significantly reduce the packet loss without any additional delay as compared to any other known technique for a OBS network. We also show the impact of reduced loss in the OBS core network on existing electronic transport protocols, such as TCP Reno. TCP over OBS with FEC/CBA achieves significantly higher throughput compared to other reliable OBS mechanism. We expect similar improvement to other TCP flavors and intend to evaluate them in the near future.

In this paper, we limit our study to static CBA, wherein the ratio of the loss-sensitive to the loss-tolerant traffic is fixed. We intend to extend the static CBA mechanism to a dynamic feedback-based CBA mechanism such that the packet ratio of the different traffic streams is dynamically adjusted based on the experienced loss (and load) along the path. Similarly we intend to extend the currently proposed static FEC to a dynamic feedback-based FEC techniques so as to add the optimal redundancy to each burst. Another area future work is to evaluate the effect of FEC/CBA schemes on recently proposed TCP flavors that can achieve better performance in a high-bandwidth and high-delay network environment.

References

[1] I. Foster, C. Kesselman, and S. Tuecker, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal of Supercomputer Applications*, vol. 15, no. 3, 2001.

[2] "GRID.IT: An Italian national research council project on grid computing funded under the national programme," *http://grid.it*.

[3] M. S. Borella, J. P. Jue, D. Banerjee, B. Ramamurthy, and B. Mukherjee, "Optical components for WDM lightwave networks," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1274–1307, August 1997.

[4] H. Zang, J.P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *SPIE Optical Networks Magazine*, vol. 1, no. 1, January 2000.

[5] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 903–908, June 1996.

[6] B. Ramamurthy and B. Mukherjee, "Wavelength conversion in WDM networking," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1061–1073, September 1998.

[7] J.S. Turner, "Terabit burst switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3–16, January 1999.

[8] 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, January 1999.

[9] J.Y. Wei, J.L. Pastor, R.S. Ramamurthy, and Y. Tsai, "Just-in-time optical burst switching for multi-wavelength networks," in *Proceedings, IFIP TC6 International Conference on Broadband Communications*, November 1999, pp. 339–352.

[10] A. Tzanakaki, I. Tomkos, D. Simeonidou, R. Nejabati, and M. J. O'Mahony, "An optical network infrastructure suitable for global grid computing," in *Proceedings, ERENA Networking Conference*, 2004.

[11] S.R. Thorpe, D.S. Stevenson, and G.K. Edwards, "Using Just-in-Time to enable optical networking for grids," in *Proceedings, Workshop on Networks for Grid Applications, BROADNETS*, October 2004.

[12] E. Van Breusegem, M. De Leenheer, and et al., "An OBS architecture for pervasive grid computing," in *Proceedings, Workshop on Optical Burst Switching, BROADNETS*, October 2004.

[13] W. Feng and P. Tinnakornsrisuphap, "The failure of TCP in high-performance computational grids," in *Proceedings, Supercomputing Conference*, 2000.

[14] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proceedings, ACM SIGCOMM*, 2002.

[15] S. Floyd, "Highspeed TCP for large congestion windows," *RFC 3649*, December 2003.

[16] R.N. Shorten and D.J. Leith, "H-TCP: TCP for high-speed and long-distance networks," in *Proceedings, 2nd International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet)*, 2004.

[17] C. Jin, D.X. Wei, and et al., "FAST TCP: From theory to experiments," *IEEE Network*, vol. 19, no. 1, pp. 4–11, January 2005.

[18] L. Brakmo and L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, October 1995.

[19] L. Brakmo, S. O'Malley, and L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proceedings, SIGCOMM,*, August 1994, pp. 24–35.

[20] E. Weigle and W.-C. Feng, "A case for TCP Vegas in high-performance computational grids," in *Proceedings, 10th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, August 2001.

[21] Y. Gu and R.L. Grossman, "UDT: An application level transport protocol for grid computing," in *Proceedings, 2nd International Workshop on Protocols for Fast Long-Distance Networks (PFLD-Net)*, 2004.

[22] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast long-distance networks," in *Proceedings, IEEE Infocom*, 2004.

[23] J.F. Kurose and K. Ross, *Computer Networking, a top down approach featuring the Internet (3rd edition)"*, Addison-Wesley Longman, 2004.

[24] V. Jacobson, "Congestion avoidance and control," in *Proceedings, ACM SIGCOMM*, 1988.

[25] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," *RFC 2001*, 1997.

[26] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," *RFC 2582*, 1999.

[27] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options," *RFC 2018*, 1996.

[28] X. Yu, C. Qiao, and Y. Liu, "TCP implementations and false time out detection in OBS networks," in *Proceedings, IEEE Infocom*, March 2004.

[29] J. Ahn, P. Danzig, Z. Liu, and L. Yan, "Evaluation of TCP Vegas: emulation and experiment," *Computer Communication Review*, vol. 25, pp. 185–95, October 1995.

[30] 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," in *Proceedings, SPIE OptiComm*, October 2000, pp. 235–246.

[31] V. M. Vokkarane and J. P. Jue, "Burst segmentation: An approach for reducing packet loss in optical burst switched networks," *SPIE Optical Networks Magazine*, vol. 4, no. 6, pp. 81–89, November-December 2003.

[32] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, and et al., "CORD: Contention resolution by delay lines," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 1014–1029, June 1996.

[33] R. Ramaswami and K.N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 489–500, October 1995.

[34] A. Bononi, G. A. Castanon, and O. K. Tonguz, "Analysis of hot-potato optical networks with wavelength conversion," *IEEE/OSA Journal of Lightwave Technology*, vol. 17, no. 4, pp. 525–534, April 1999.

[35] F. Forghieri, A. Bononi, and P. R. Prucnal, "Analysis and comparison of hot-potato and single-buffer deflection routing in very high bit rate optical mesh networks," *IEEE Transactions on Communications*, vol. 43, no. 1, pp. 88–98, January 1995.

[36] G.P.V. Thodime, V. M. Vokkarane, and J. P. Jue, "Dynamic congestion-based load balanced routing in optical burst-switched networks," in *Proceedings, IEEE Globecom*, December 2003, vol. 5, pp. 2694–2698.

[37] J. Li, G. Mohan, and K. C. Chua, "Load balancing using adaptive alternate routing in IP-over-WDM optical burst switching networks," in *Proceedings, SPIE OptiComm*, October 2003.

[38] B. Chen and J. Wang, "Hybrid switching and p-routing for optical burst switching networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1071–1080, September 2003.

[39] L. Yang and G.N. Rouskas, "Path switching in obs networks," in *Proceedings, Networking 2005*, May 2005.

[40] J. Li and C. Qiao, "Schedule burst proactively for optical burst switching networks," in *Proceedings, IEEE Globecom*, December 2003, pp. 2787–2791.

[41] Q. Zhang, V. M. Vokkarane, Y. Wang, and J. P. Jue, "Evaluation of burst retransmission in optical burst-switched networks," in *Proceedings, IEEE Broadnets 2005, Optical Networking Symposium*, October 2005.

[42] X. Huang, V.M. Vokkarane, and J.P. Jue, "Burst cloning: A proactive scheme to reduce data loss in optical burst-switched networks," in *Proceedings, IEEE International Conference on Communications (ICC)*, May 2005.

[43] D. Griffith and S. Lee, "A 1 + 1 protection architecture for optical burst switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 9, 2003.

[44] S.R. Murthy, P. Jayachandran, and P. Bhamidipati, "On using forward error correction to provide protection in optical burst switched networks," in *Proceedings, IEEE Workshop on High Performance Switching and Routing (HPSR)*, May 2005.

[45] T. Mizuochi and et.al, "Forward error correction based on block turbo code with 3-bit soft decision for 10-gb/s optical communication systems," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 10, no. 2, pp. 376–386, Mar. 2004.

[46] V. M. Vokkarane and J. P. Jue, "Prioritized burst segmentation and composite burst assembly techniques for QoS support in optical burst switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1198–1209, September 2003.

[47] V. M. Vokkarane and J. P. Jue, "Segmentation-based non-preemptive scheduling algorithms for optical burst-switched networks," in *Proceedings, First International Workshop on Optical Burst Switching (WOBS), co-located with OptiComm 2003*, October 2003.

[48] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, April 2000.