MIT Open Access Articles

A Joint Design of Congestion Control and Burst Contention Resolution for Optical Burst Switching Networks

Massachusetts Institute of Technology

A Joint Design of Congestion Control and Burst Contention Resolution for Optical Burst Switching Networks

Won-Seok Park, Minsu Shin, Hyang-Won Lee, *Associate Member, IEEE*, and Song Chong, *Member, IEEE*

Abstract—This paper revisits burst contention resolution problems in optical burst switching (OBS) networks from the viewpoint of network utility maximization. Burst collision occurs when two or more bursts access the same wavelength simultaneously, and the occurrence becomes more frequent as the offered load increases. In particular, when the network is overloaded, no contention resolution scheme would effectively avoid the collision without the help of congestion control. We formulate a joint optimization problem where two variables, the length and the time at which each burst is injected into the network, are jointly optimized in order to maximize aggregate utility while minimizing burst loss. A distributed algorithm is also developed, which explicitly reveals how burst contention resolution and congestion control must interact. The simulation results show that the joint control decouples throughput performance from burst loss performance so that burst loss ratio does not increase as network throughput increases. This is not the case in conventional contention resolution schemes where burst loss ratio increases as network throughput increases so that achievable network throughput is limited. Our work is the first attempt to the joint design of congestion and contention control and might lead to an interesting development in OBS research.

Index Terms—Efficiency, fairness, flow control, lightpath, optical burst switching, proportional fairness.

I. INTRODUCTION

OPTICAL burst switching (OBS) [1]–[7] has been proposed as a future high-speed switching technology that compromises the pros and cons of optical circuit switching and optical packet switching. In OBS networks, multiple IP packets with the same destination are assembled into a burst at an ingress OBS node and the burst is transmitted through the network core entirely in the optical domain. The ingress OBS node also sends a corresponding control packet (or burst header packet) for each

data burst on a separate control channel. Because this control packet leads the data burst by an offset time and reserves wavelength resources for its data burst in advance, OBS can eliminate the need for buffering of the data burst unlike conventional IP network which can buffer its data packet while processing the corresponding header and configuring its switching path.

Although OBS has been considered as a promising solution for optical networks due to its bandwidth efficiency over optical circuit switching and its implementation simplicity over optical packet switching, it has an unsolved technical challenge known as burst contention resolution problem. Since most of the current optical core nodes are not supporting optical buffers, or if any, with highly limited storage capacity, burst contention is unavoidable when two or more bursts are competing for the same wavelength resource at the same time at the same link. Hence, various contention resolution schemes have been proposed using time deflection [5], space deflection [6], and wavelength conversion [1], [7], [8]. Unfortunately, resolving the burst contention based solely on these reactive schemes has a fundamental limitation in that they might be effective when the network is underloaded but no longer effective as the network becomes congested. An alternative approach in this regard is to proactively prevent an OBS network from entering the congestion state which may incur lots of burst contentions. Some recent schemes [4], [9] including our earlier work are in this line of work adopting congestion control approach [10]–[14] in electrical packet switching networks. Especially, our previous work [9] investigated the impact of flow control and burst delay control on the performance of OBS network. However, those two controls work independently of each other, and hence, it was not clearly understood how they have to interact with each other in order to improve the performance of OBS network.

Obviously, congestion control cannot solely solve the entire problem. The possibility of burst contention always exists even in the lightly loaded cases as long as two or more bursts demand the same wavelength at the same time at the same link. In these cases, one of them should be dropped even though there is sufficient leftover bandwidth. In electrical packet switching networks, such a loss can be effectively mitigated by using buffers in conjunction with congestion control at the edge or source. In the optical domain, however, buffering is still an immature technology in terms of cost and capacity [15]. Thus, an extra effort that we can consider in conjunction with congestion control is to schedule the injection times of bursts at the ingress edge in a way that no burst contention occurs inside the network. The main theme of this paper is to exploit this interaction between

W.-S. Park is with LG Electronics Inc., Gumchon-gu, Seoul 153-802, Korea (e-mail: pwsspm1@gmail.com).

M. Shin is with SK Broadband, Jung-gu, Seoul 100-711, Korea (e-mail: minsu.shin@gmail.com).

H.-W. Lee is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: hwlee@mit.edu).

S. Chong is with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: song@ee.kaist.ac.kr).

congestion control and burst transmission scheduling and their joint design to maximize the network throughput while minimizing burst losses and ensuring a ceratin fairness among burst flows such as proportional fairness [10].

We consider a network utility maximization problem with link capacity constraints and burst non-overlapping constraints, where two variables to be jointly optimized are burst length (congestion control variable) and offset time (burst contention resolution variable). Both variables can be easily adjusted at the ingress edge by a timer-based burst assembler with an electrical buffer [1]. A distributed algorithm to solve this problem is developed assuming explicit signaling between links and edges using OBS control channels. The significance of the proposed joint control is that it decouples throughput performance from burst loss performance so that one can load the network up to almost its capacity with virtually no burst losses in an asymptotic sense. A disclaimer is that in some dynamic scenarios where, for instance, edge-to-edge burst flows frequently joint and leave, transient burst losses are inevitable and sometimes non-negligible even with our proposed joint control. However, such burst losses do not persist under our joint control. We also show through simulations that transient losses can be substantially reduced by lowering the target utilization of congestion control part of our algorithm. To our best knowledge, this work is not only the first attempt to the joint design of congestion and contention control for OBS networks but also the first report showing that throughput performance can be decoupled from burst loss performance.

The rest of this paper is organized as follows. In Section II, we provide a detailed description of OBS network model. In Section III, we define a network utility maximization problem with link capacity and non-overlapping constraints. In Section IV, we propose a joint congestion control and burst contention resolution algorithm based on the network utility maximization problem. Simulation results and discussions are presented in Section V, and we conclude the paper in Section VI.

II. OBS NETWORK MODEL

We consider an OBS network which consists of a set of fiber links, $L = \{1, \ldots, L\}$, where every fiber link $l \in L$ has a set of channels with different wavelengths. We assume that the capacity of each wavelength is equal to C in the network. The network resources are shared by a set of edge-to-edge optical burst flows, $S = \{1, \ldots, S\}$, each of which is assumed to have a predefined single route between its ingress edge and egress edge, denoted by a set $L(s) \subset L$. We assume that no wavelength converter resides in the network due to its cost and complexity [8]. Hence each flow uses the same wavelength in its route. Therefore, each wavelength plane of the network can be modeled separately. For this reason, we consider a single wavelength plane of the network solely without loss of generality. Let $S(l) \subset S$ be the set of flows whose routes include link l.

We assume a timer-based burst assembler [1] so that each flow s generates one burst in every τ second interval. The burst assembly interval τ is assumed to be fixed and common for all flows in the network. The parameter τ cannot be arbitrarily large because it will increase edge-to-edge (and hence end-to-end)
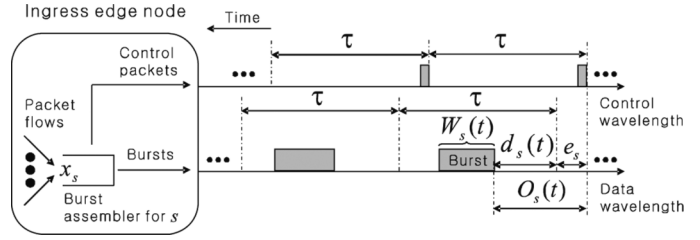

Fig. 1. The time diagram for departure of both control packets and bursts of the flow $s \in S$ at ingress edge node: The burst with its duration $W_s(t)$ (sec) and its control packets are sent every burst assembly time τ (sec).

delay. On the other hand, it should be larger than (multiple of) the packet processing time at an edge node to ensure that a single burst can contain multiple packets. Hence, the value τ cannot be selected arbitrarily but should be selected from between multiples (determined by certain design criterion) of packet processing time and delay requirement. Let b_s be the variable specifying the allowed burst length (in seconds) and W_s be the actual burst length (in seconds) being injected into the network. Obviously, we have $0 \leq W_s \leq b_s \leq \tau$, and $W_s = b_s$ if there is sufficient data backlog and W_s can be zero if there is no data backlog.

As for the channel reservation, we assume one-way reservation protocol called *Just-Enough-Time* [1]. As shown in Fig. 1, source s having a data burst to transmit first sends a control packet to its egress edge along its route $L(s)$ but using a separate signaling channel. After O_s seconds, the source transmits the data burst. So, the burst follows the control packet which reserves channels for the burst in advance along the route. We call O_s as offset time. The control packet carries two values, offset time O_s indicating when the burst will arrive and actual burst length W_s indicating how long the burst will be. The offset time O_s of the flow s consists of fixed base offset time e_s for compensating its control packet's processing time along its lightpath and variable extra offset time d_s for other purpose (e.g., contention resolution in this paper).

Let x_s (bytes/sec) and d_s (in seconds) be the allowed burst assembling rate (allowed edge-to-edge aggregated data transmission rate) and variable offset time of source (edge-to-edge burst flow) s respectively, and τ be the period of burst assembly as shown in Fig. 1. Then, the allowed burst length b_s of s (in seconds) can be expressed as $b_s = (x_s \tau)/(C)$ where C (bits/sec) is the single wavelength capacity. Let t_{sl}^{st} and t_{sl}^{end} denote the start time and end time of a burst at link l, respectively. We also assume that the guard time g is used between consecutive bursts to compensate the configuration and switching time of core node and to avoid a burst contention. So, if the arrival time of control packet from source s at link l is defined as T_{sl}, then the two variables can be written as $t_{sl}^{st} = T_{sl} + O_s (= e_s + d_s)$ and $t_{sl}^{end} = t_{sl}^{st} + b_s + g$.

The sources $S(l)$ are ordered according to the arrival times of their control packets. This order is fixed because the arrival times are determined by both the time-invariant propagation delay and the start time of its control packet at the ingress edge. Formally, we write $S(l) = \{s_l^1, s_l^2, \ldots, s_l^{N_l}\}$ where s_l^i denotes the source whose control packet arrives at link l at the ith earliest time of all the control packets traversing the same wavelength
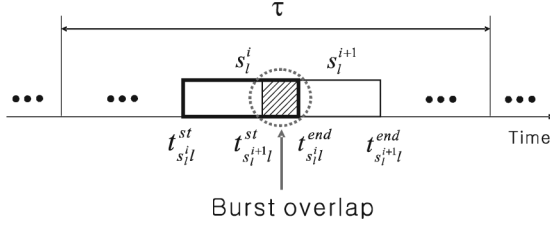
Fig. 2. An example of burst overlap: In a single wavelength in link l, burst overlap occurs when the end time of a burst is later than the start time of its consecutive burst. The burst flow sources denoted by s_l^i and s_l^{i+1} are named after the order of arrival times of their control packets.

at link l, and N_l is the cardinality of $S(l)$, i.e., $N_l = |S(l)|$. Based on this, the burst overlap can be modelled as shown in Fig. 2 where the burst overlap between burst flows s_l^i and s_l^{i+1} is represented by the period $[t_{s_l^{i+1}l}^{\text{st}}, t_{s_l^i l}^{\text{end}}]$. To avoid the overlap, it must hold $t_{s_l^i l}^{\text{end}} \leq t_{s_l^{i+1}l}^{\text{st}}$.

Each source s is associated with utility function $U_s(b_s)$ of its burst length b_s, which is assumed to be continuously differentiable, increasing and strictly concave.

III. PROBLEM FORMULATION

For the greedy burst assembler where b_s is set to τ, the actual burst length W_s can grow up to τ when there is sufficient backlog, which is obviously an unacceptable situation since any burst which shares a link with flow s would collide. Thus, it is necessary to have a mechanism to limit b_s and control d_s such that the resource τ is fairly and efficiently shared by contending flows without burst overlap at every link. To do this, we will propose a joint congestion control and burst contention resolution algorithm by using the following problem ($\mathbf{P}$):

$$\max_{b,d} \quad \sum_{s \in S} U_s(b_s) \tag{1}$$

$$\text{subject to} \quad \sum_{s \in S(l)} b_s \leq \tau, \forall l \tag{2}$$

$$t_{s_l^i l}^{\text{end}} \leq t_{s_l^{i+1}l}^{\text{st}}, i = 1, \ldots, N_l, \forall l \tag{3}$$

$$m_s \leq b_s \leq M_s, d_s \geq 0, \forall s \tag{4}$$

where $b = [b_s, s \in S], d = [d_s, s \in S]$. Equation (2) is the link capacity constraint which means that the aggregate bandwidth usage by burst flows must be less than or equal to the capacity of a wavelength at each link at anytime. We formally represent this constraint as follows: for each link $l \in L, \sum_{s \in S(l)} x_s \leq C$, or equivalently, $\sum_{s \in S(l)} b_s \leq \tau$. Equation (3) is the burst no-overlap constraint which guarantees collision-free burst arrivals at steady state. So, it is easy to see that the solution (b^*, d^*) to the problem ($\mathbf{P}$) maximizes the sum of utility functions while ensuring that the sum of the burst lengths at a wavelength of each link does not exceed the period τ and there is no burst overlap (or loss) at steady state. In (4), m_s and M_s respectively

denote the minimum required burst length and the maximum possible burst length of source s, where $0 \leq m_s \leq M_s \leq \tau$. Each edge-to-edge flow s serves the end-to-end flows routed on the path of s. Hence, the minimum requirement m_s will be determined based on the estimated end-to-end traffic demands over edge-to-edge flow s. These minimum requirements, however, cannot exceed the link capacities, i.e., $\sum_l m_s < C$, for all links l. Hence, if the network cannot support the minimum requirements, they will have to be adjusted in order to satisfy the link capacity constraints (e.g., by normalizing). Once these feasible m_s's are fixed, our flow control guarantees m_s for each edge-to-edge flow s and distributes the leftover capacity (remaining after minimum guarantee) in proportional fair sense.

IV. PROPOSED ALGORITHM

If the primal objective function is strictly concave with respect to (b, d) and the feasible region is convex, then the problem can be easily solved by using duality theory and gradient projection method [16]. However, the primal objective function in (1) is strictly concave in b but not in (b, d). Consequently, we cannot apply gradient projection method to the dual problem because when the primal objective function is not strictly concave in maximization problem, its dual objective function is not differentiable in general. One may decompose the problem into two subproblems, including the one with respect to b and the one with respect to d, and apply duality theory and subgradient projection method to the former and linear programming method to the latter. There are several linear optimization methods that can generate optimal solutions reasonably fast [17], but unfortunately, it is hard to implement those methods in a distributed way, which implies that they are not suitable for the development of network algorithm.

A. Augmented Lagrangian Method

In order to circumvent this difficulty and develop a distributed algorithm, we use augmented Lagrangian method, which is regarded as one of the penalty function methods [18]. Consider the dual problem

$$\min_{\lambda \geq 0, \mu} D(\lambda, \mu) \tag{5}$$

where $D(\lambda, \mu)$ is the dual objective function defined as

$$D(\lambda, \mu) = \max_{b,d,z} L_A(b, d, z, \lambda, \mu), \tag{6}$$

where $L_A(\cdot)$ is defined below. After converting the inequality constraints in (3) to equality constraints using the additional quadratic variables $z_{il}^2, i = 1, \ldots, N_l, \forall l \in L$, the objective function in (1) is augmented as follows:

$$f(b, d, z) \triangleq \sum_{s \in S} U_s(b_s) - \frac{1}{2}\kappa \sum_l \sum_{i=1}^{N_l} \left(t_{s_l^i l}^{\text{end}} - t_{s_l^{i+1}l}^{\text{st}} + z_{il}^2 \right)^2 \tag{7}$$

where κ is a positive penalty parameter. Then, the augmented Lagrangian function is defined as

$$L_A(b,d,z,\lambda,\mu) = f(b,d,z) + \sum_l \lambda_l \left(\tau - \sum_{s \in S(l)} b_s \right)$$
$$- \sum_l \sum_{i=1}^{N_l} \mu_{il} \left(t_{s_l^i l}^{\mathrm{end}} - t_{s_l^{i+1} l}^{\mathrm{st}} + z_{il}^2 \right) \quad (8)$$

where $\lambda = [\lambda_l, \forall l \in L]$ and $\mu = [\mu_{il}, i = 1, \ldots N_l, \forall l \in L]$ are Lagrangian multiplier vectors, and $z = [z_{il}, i = 1, \ldots, N_l, \forall l \in L]$ is the slack variable which is used to convert (3) into equality constraints. The augmented Lagrangian function $L_A(\cdot)$ can also be considered as a usual Lagrangian function augmented by a penalty term. Let $m = [m_s, \forall s \in S]$ and $M = [M_s, \forall s \in S]$. By the method of multipliers [16], we have the following successive maximization of the form:

$$(b(t+1), d(t+1))$$
$$= \arg_{(b,d)} \max_{m \leq b \leq M, d \geq 0, z} L_A(b,d,z,\lambda(t),\mu(t)) \quad (9)$$

followed by updates of the vectors $\lambda(t)$ and $\mu(t)$ according to

$$\lambda_l(t+1) = \left[\lambda_l(t) - \gamma \left(\tau - \sum_{s \in S(l)} b_s(t) \right) \right]^+ \quad (10)$$

$$\mu_{il}(t+1) = \left[\mu_{il}(t) + \kappa \left(t_{s_l^i l}^{\mathrm{end}}(t) - t_{s_l^{i+1} l}^{\mathrm{st}}(t) \right) \right]^+ \quad (11)$$

where $[\cdot]^+$ denotes non-negativity projection and γ is a positive step size. The maximization in (9) at each time t is separable and can be solved by the following iterations:

$$b_s(t) = \left[U_s'^{-1} \left(p_s^b(t) \right) \right]_{m_s}^{M_s}$$
$$p_s^b(t) = \sum_{l \in L(s)} \{ \lambda_l(t) + \mu_{I(s,l)l}(t) \}, \quad (12)$$

and

$$d_s(t+1) = \left[d_s(t) - \kappa p_s^d(t) \right]^+$$
$$p_s^d(t) = \sum_{l \in L(s)} \{ \mu_{I(s,l)l}(t) - \mu_{\bar{I}(s,l)l}(t) \} \quad (13)$$

where $[\cdot]_{m_s}^{M_s}$ denotes the projection onto the interval $[m_s, M_s]$, $L(s)$ is the set of links which flow s traverses, $I(s,l)$ indicates the order of source s at a wavelength of link l and $\bar{I}(s,l) = I(s,l) - 1$. So if the control packet's arrival time of s is the ith earliest at link l, then $I(s,l) = i$ and $\bar{I}(s,l) = i - 1$. In (13), $\mu_{0l}(t)$ is defined to be zero for every t. The detail of the derivation is given in Appendix I.

B. Algorithm and Its Interpretation

Using the results (10)–(13), we propose a joint flow and burst offset time control algorithm to maximize network utility while achieving zero burst loss. For better understanding, we first show the overall architecture of our algorithm (see Fig. 3). The ingress edge node A assembles multiple IP packets (destined to the egress node B) into a burst. Before this burst is sent to the destination egress node B, a forward control packet (FCP)
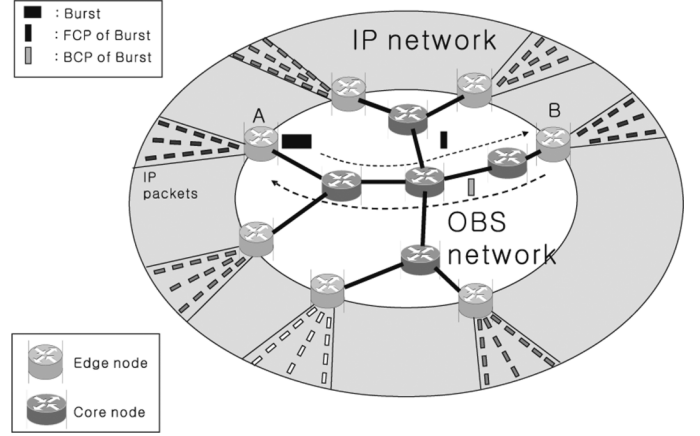

Fig. 3. The conceptual architecture of OBS network: The ingress edge node A, which makes a burst out of outer IP packets, sends the corresponding FCP (Forward Control Packet) of the burst and, the offset time after, it sends the burst to the egress node B. Then, the node B sends the RCP (Reverse direction Control Packet) back to the node A when it receives the FCP.

is sent to reserve the resource for this burst. This FCP also collects the congestion price (λ_l) and contention price (μ_{il}) along its paths, and a reverse-direction control packet (RCP) sent by egress node delivers this information to the engress node. The ingress node uses this information (p_s^b and p_s^d) to update its burst length and offset time.

Algorithm 1:
- Ingress Edge (for $s \in S$)
  — At every time t,
    — update the burst offset time by
    $$d_s(t+1) = \left[d_s(t) - \kappa p_s^d(t) \right]^+ \quad (14)$$
    — update the burst length by
    $$= b_s(t) \left[U_s'^{-1} \left(p_s^b(t) \right) \right]_{m_s}^{M_s} \quad (15)$$
    $$\left(b_s(t+1) = \left[b_s(t) - \nu \left(p_s^b(t) - U_s(b_s(t))' \right) \right]_{m_s}^{M_s} \right) \quad (16)$$
  — Upon every τ seconds,
    — generate and transmit an FCP containing t_{sl}^{st}, t_{sl}^{end} and b_s
    — transmit a data burst $O_s(= e_s + d_s)$ seconds after the FCP was sent
  — Upon reception of a RCP,
    — update p_s^d and p_s^b
- Egress Edge (for $s \in S$)
  — Upon reception of an FCP,
    — send a RCP containing p_s^b and p_s^d to its ingress edge node
- *Link l*
  — At every time t,
    — update congestion price by
    $$\lambda_l(t+1) = \left[\lambda_l(t) - \gamma \left(\tau - \sum_{s \in S(l)} b_s(t) \right) \right]^+ \quad (17)$$
    — update burst overlap prices for all $i \in S(l)$ by
    $$\mu_{il}(t+1) = \left[\mu_{il}(t) + \kappa \left(t_{s_l^i l}^{\mathrm{end}}(t) - t_{s_l^{i+1} l}^{\mathrm{st}}(t) \right) \right]^+ \quad (18)$$

— Upon reception of an FCP of flow s,
  — update t_{sl}^{st} and t_{sl}^{end}
  — update p_s^b and p_s^d in the FCP and send it to the next hop.

As is well known, λ_l is interpreted as the price for link congestion. As seen in (17), if link l is congested, then λ_l will increase, and otherwise, it will decrease. For source s, the sum of these λ_l's on its path is delivered to the source, and used in burst length adaptation (15). Since the utility function is assumed to be increasing and strictly concave, source s will increase its burst length b_s if the sum decreases, and otherwise, it will decrease b_s. Thus, λ_l is well interpreted as link congestion price. The smoothing of the burst length can be performed by replacing the update (15) by (16) where ν is a positive step size. If convergent, this update will find a point satisfying $p_s^b(t) = U_s(b_s(t))'$, which is equivalent to (15). This will effectively avoid the fluctuation of the burst length; however, it will take longer to reach the steady state.

Similarly, μ_{il} is interpreted as the price for burst overlap. As seen in (18), if the ith and $(i+1)$th bursts overlap, then μ_{il} will increase. Otherwise, it will decrease. The sum of these $\mu_{I(s,l)l}$'s divided by τ on its path is delivered to source s, which adapts its burst length according to (15). Thus, any overlap event associated with the burst of s will result in the decrease of b_s. On the other hand, if there is no overlap of burst on its path, b_s could increase depending on the sum of λ_l's on its path. Let us see how μ_{il} affects the adaptation of burst offset time d_s. Consider source s and link l, and suppose $I(s,l) = i$. Then, μ_{il}, which is the price associated with the burst from s and the $(i+1)$th burst arriving at link l, contributes to the increase of p_s^d. This in turn leads to the decrease of offset time d_s, which is reasonable because burst s should be sent earlier to avoid the overlap. In contrast, if burst s overlaps with the $(i-1)$ burst, increased $\mu_{(i-1)l}$ will result in the increase of offset time d_s. As the number of edge-to-edge flows increases, the computational complexity of contention price will linearly increase, because it is computed between neighboring bursts. In contrast, the congestion price is computed based on aggregate flow at each link, and thus, it will not be affected by the number of flows. Note that our control is based on edge-to-edge flows, and its number is relatively low. Hence, the scalability with respect to the number of flows may not be a concern.

C. Convergence

For the convergence of gradient-type iteration, there is a standard technique that gives the condition on convergent step size such that the iteration converges to the optimal solution, even in the presence of communication delays [12], [19]. However, the standard technique cannot be applied here because the dual objective function is not differentiable. In fact, the synchronous version of the iterations (14)–(18) converges to within some range of the optimal value as shown in the following theorem. Let $p = [\lambda^T, \mu^T]^T$ and $v(p)$ be the subgradient vector of the dual function D at point p.

Theorem 4.1: Suppose that there exists a constant $V > 0$ such that $\|v(p(t))\|_2 \leq V$ for all t. Then, the algorithm (10)–(13) converges to within $(\gamma V^2)/(2)$ of optimal.

Proof: The proof is straightforward following Theorem 4.1 in [20] and omitted for brevity.

The above theorem shows that the smaller the step size is taken, the closer the algorithm converges to the optimal. $v(p)$ is given as the vector of (19) and (20), and it is clear that $\|v(p)\|_2$ is bounded.

V. SIMULATION RESULTS

In this section, we present some simulation results verifying the performance of the proposed joint control algorithm in various optical network topologies. First we examine the convergence property of the OBS networks when the proposed scheme is used. The results show that after some transient period the system enters steady-state where the amount of burst loss converges to zero. Next, we compare the performance of the joint control algorithm with conventional burst contention resolution schemes. We adopt two performance metrics for the comparison. One is burst loss ratio, which is the ratio of the number of lost bursts to the total number of transmitted bursts. The other is goodput, the amount of bursts per second that reached egress edge successfully. Obviously, lower burst loss ratio and higher goodput represent better efficiency in the OBS network. In the comparative simulation, we can demonstrate that the proposed algorithm outperforms the conventional algorithms. The conventional scheme generates a burst by aggregating all the packets that have arrived until the burst timer is expired, which is different from our algorithm that assembles and transmits the burst following to the controlled burst length information. Below, we introduce the general contention-resolution schemes that we use in simulation.

1) Randomized offset time [3]: To prevent repetitive collision when using timer-based burst assembler, the extra offset time $d(t)$ in Fig. 1 is randomized only when the collision is detected at ingress edge node. At every τ, we update $d_s(t+1) = [0, \tau - W_s(t)]^+ \cdot u(0,1)$ if burst loss is reported by the control packet and $d_s(t+1) = d_s(t)$ otherwise, where $u(0,1)$ is a uniformly distributed random number in [0,1].

2) Fiber Delay Lines (FDL) [2], [5]: FDL provides optical buffering capability to core nodes. The potential contentions among bursts that slightly overlap can be resolved efficiently using FDL. In some cases, we set up per port FDLs with its delay unit $D = \tau/100$, and vary its buffer size N for maximum delay ND. Apparently, if the size N is larger, more contentions can be resolved. We will show that the proposed algorithm outperforms those conventional algorithms even without FDL.

A. Simulation Environment

The simulation is performed in the NS-2 [21] environment. We implement the proposed algorithm in Section IV at ingress/egress edge and core optical node models. The congestion and burst overlap prices are conveyed by the control packets including FCP and RCP. Since our algorithm can be separately implemented over any wavelength resource, only a single wavelength is used in the simulation.

We set the timer value $\tau = 100$ μs and burst guard time $g = 1$ μs at every ingress edge node, and the single wavelength capacity $C = 10$ Gbps. We also set the basic offset time equally as $e_s = 50$ μs. Because the control packet is processed at every
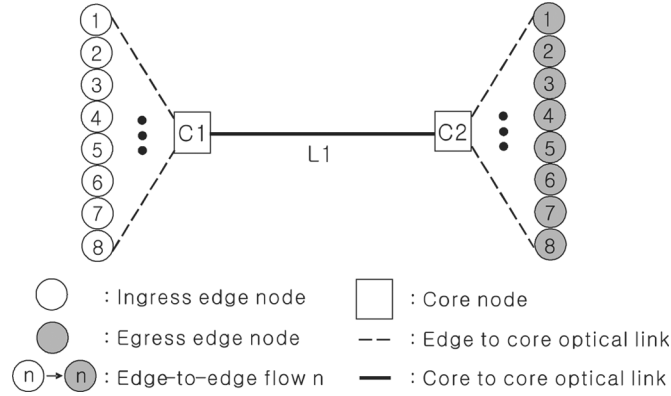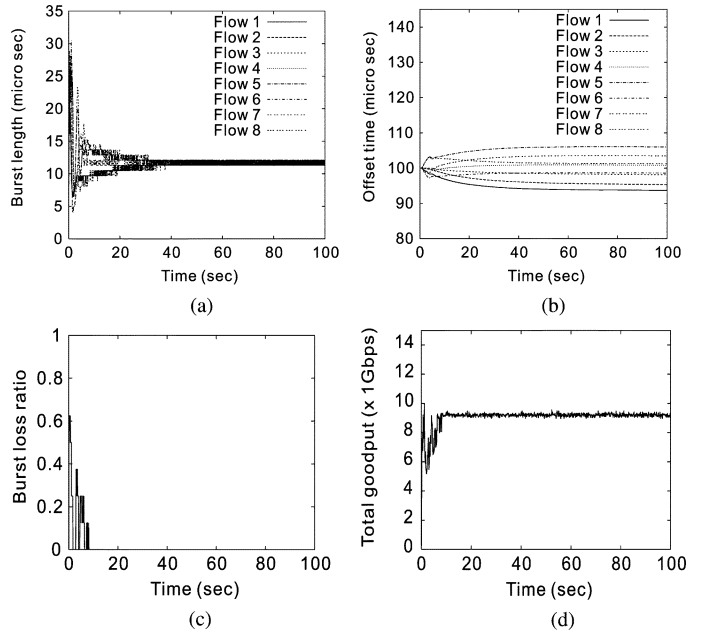
Fig. 4. The dumbbell topology.

Fig. 5. Network convergence in the single-link scenario: eight burst flows are sharing a single bottleneck link. After some transient time, the network converges to the ideal state of zero loss and maximum goodput. (a) Burst length b_s. (b) Offset time O_s. (c) Burst loss ratio. (d) Total goodput.

node it passes, e_s must account for the processing time at all the nodes on the path of edge-to-edge flow s. Otherwise, a data burst can pass its associated control packet. The additional offset time d_s is initially set to 50 μs and can be adjusted within the range of [0, 100] μs. The minimum and maximum burst lengths are fixed as $m_s = 2\ \mu s$ and $M_s = 50\ \mu s$, respectively.

Each edge-to-edge burst flow $s \in S$ is composed of 40 on-off traffic sources which follow Pareto distribution with shaping factor (tail index) 1.2 [22]. The packet size is fixed to 1000 Bytes. We use logarithm utility function $U_s(b_s) = \log(b_s)$ for each flow s to achieve proportional fairness [10], which is believed to be a good compromise between fairness and efficiency.

Remark: It was shown in [22] that the aggregate TCP traffic follows the Pareto distribution, if the file object sizes are assumed to be Pareto distributed. Moreover, the Internet traffic is self similar, and this self similarity can be effectively modeled by Pareto distribution. Therefore, we believe that our simulation using the Pareto traffic somehow predicts the performance of TCP. We discuss more on TCP in the next section.

B. Convergence in Dumbbell Topology

First, we test the convergence of our algorithm in the dumbbell topology shown in Fig. 4. Eight edge-to-edge burst flows share a single link L1 and have randomly selected edge-to-edge propagation delays within [33, 36] ms [23]. In this scenario, the equal share of optical link capacity is an optimal fair share. All the burst lengths shown in Fig. 5(a) converge to the same value after transient period. Fig. 5(b) plots the offset time of each burst flow, and we can see that they converge to different values. The more interesting results are depicted in Fig. 5(c) and (d). As seen in Fig. 5(c), the burst loss ratio eventually converges to zero, which implies that the burst contention can be perfectly resolved by our algorithm at steady state. Furthermore, observe from Fig. 5(d) that the bottleneck link is utilized to the maximum available level except for the inter-burst guard time. This result is not easily attained by using only flow control or contention resolution. Hence, this shows the necessity that the burst length and its placement are controlled at the same time as our proposed method.

We further examine our algorithm in the dynamic scenario where the burst flows join or leave the network over time. Table I

TABLE I
THEORETICAL FAIR BURST LENGTHS IN THE DYNAMIC
SINGLE-LINK SCENARIO

s	Join (sec)	Leave (sec)	Fair $b_s(\mu sec)$				
			0~50	50~100	100~150	150~200	200~250
1	0	∞	24	15.67	11.5	15.67	24
2	0	∞	24	15.67	11.5	15.67	24
3	0	∞	24	15.67	11.5	15.67	24
4	0	∞	24	15.67	11.5	15.67	24
5	50	200	–	15.67	11.5	15.67	–
6	50	200	–	15.67	11.5	15.67	–
7	100	150	–	–	11.5	–	–
8	100	150	–	–	11.5	–	–

describes the arrival and departure time of each flow and the theoretical fair values corresponding to the scenario. Fig. 6(a) shows that the burst length of each burst flow follows the theoretical fair value after the transient period. We can see from Fig. 6(b) and (c) that our algorithm finds an arrangement of bursts in such a way that the burst loss is suppressed to zero whenever the system gets out of transient period and enters steady state. A similar result can be observed in Fig. 6(d) where the link utilization drops upon arrival or departure and then bounces back to its maximum available value (target link utilization 0.92): this target link utilization can be controlled by replacing τ in the (2) with $\alpha\tau(0 \leq \alpha \leq 1)$ and adjusting the value of α. One possible approach to suppress the transient loss is to reduce the target link utilization. So far we try to find the solutions that satisfy maximum available utilization except for the guard time of each flow. Certainly, if we increase the inter-burst guard time between bursts, the transient burst loss can be reduced significantly, of course at the cost of decreased utilization. Fig. 7 shows the result when the link target utilization is
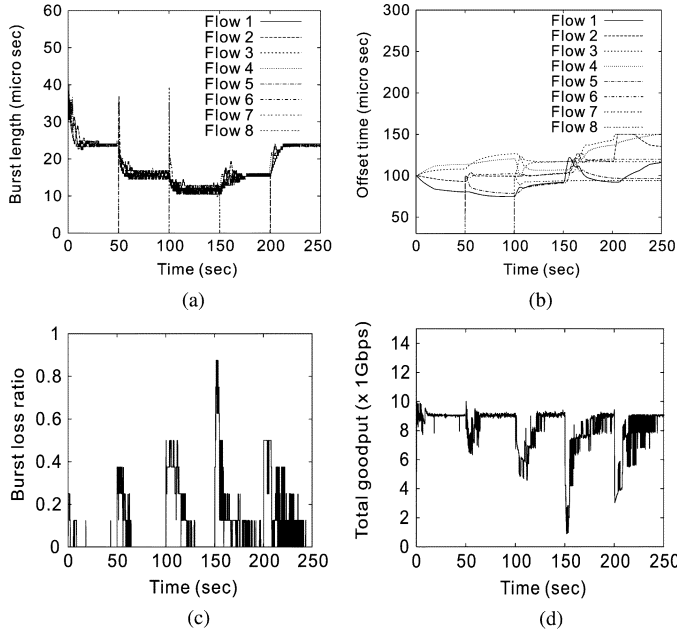
Fig. 6. Simulation results with dynamic arrival/departure in the single-link scenario: the loss ratio and goodput performance is degraded temporarily during a transient period, but it is recovered when the network enters steady state. (a) Burst length b_s. (b) Offset time O_s. (c) Burst loss ratio. (d) Total goodput.
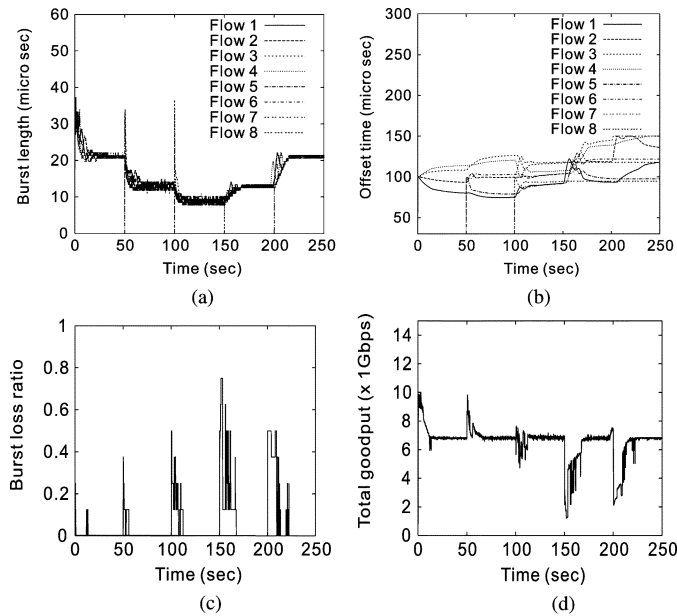

Fig. 7. Simulation results with dynamic arrival/departure in the single-link scenario (target utilization 0.7 and guard time 3.0 μs): the burst loss performance degradation in transient period is substantially reduced by the cost of lowering utilization. (a) Burst length b_s. (b) Offset time O_s. (c) Burst loss ratio. (d) Total goodput.

0.7 and the guard time g is 3 μs. We can get better transient performance than that of the maximal link utilization case.

These results confirm that the proposed algorithm finds an optimal arrangement and lengths of bursts to achieve no burst loss and high link utilization even when the burst flows dynamically join or leave the network.
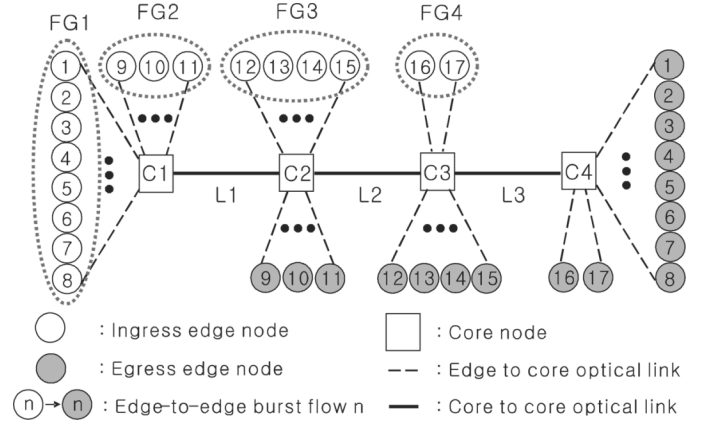

Fig. 8. The parking lot topology for multiple-link scenario.

C. Convergence in Parking Lot Topology

We also simulate the multiple-link scenario as shown in Fig. 8 where multiple core links can be bottleneck and 17 burst flows exist. The flow group 1 (FG1) consisting of eight burst flows traverses links L1, L2, and L3, the FG2 of three burst flows traverses link L1 only, the FG3 of four burst flows traverses link L2 only, and the FG4 of two burst flows traverses link L3 only. The propagation delays between the core nodes are equally fixed to 10 ms, and those between the edge and core nodes are randomly selected within [7, 13] ms [23].

Fig. 9(a)–(d) shows the convergence of each burst length. We can see that the flows traversing more hops achieve lower burst length, e.g., the flows in FG1 yield about 6 μs burst while as those in FG2 yield about 15 μs. This is due to the proportional fairness which discriminates against the flows traversing more hops. Fig. 9(e) and (f) shows that the offset time of each burst converges to a different value to avoid burst collision. In brief, the proposed algorithm works as designed in the multiple-link scenario as well as in the single-link scenario.

D. Performance Comparison

We demonstrate how much our proposed algorithm enhances the performance (i.e., burst loss ratio and total goodput) compared with those of the conventional contention resolution schemes in various topologies. In addition, we show that our algorithm also outperforms randomized offset time control scheme in edge-to-edge burst delay.

The simulations are performed at three different topologies including the dumbbell network in Fig. 4, multiple-link network in Fig. 8, and 14-node NSF mesh network in Fig. 11(a). The network like NSF topology is likely to experience serious congestion problem as the offered load increases. In order to see how the proposed and conventional algorithms react to the congestion, their performances are compared while increasing the aggregate input traffic rate into each ingress node. We take $1Q$ (≈ 0.24 Gbps) as the amount of edge-to-edge input traffic increment by each step in the single- and multiple-link topology, and $1\bar{Q}$ (≈ 0.47 Gbps) in the NSF topology.

Fig. 10 plots the burst loss ratio and goodput performance of each algorithm as the offered load increases. The performance comparison in the single-link scenario is shown in Fig. 10(a),
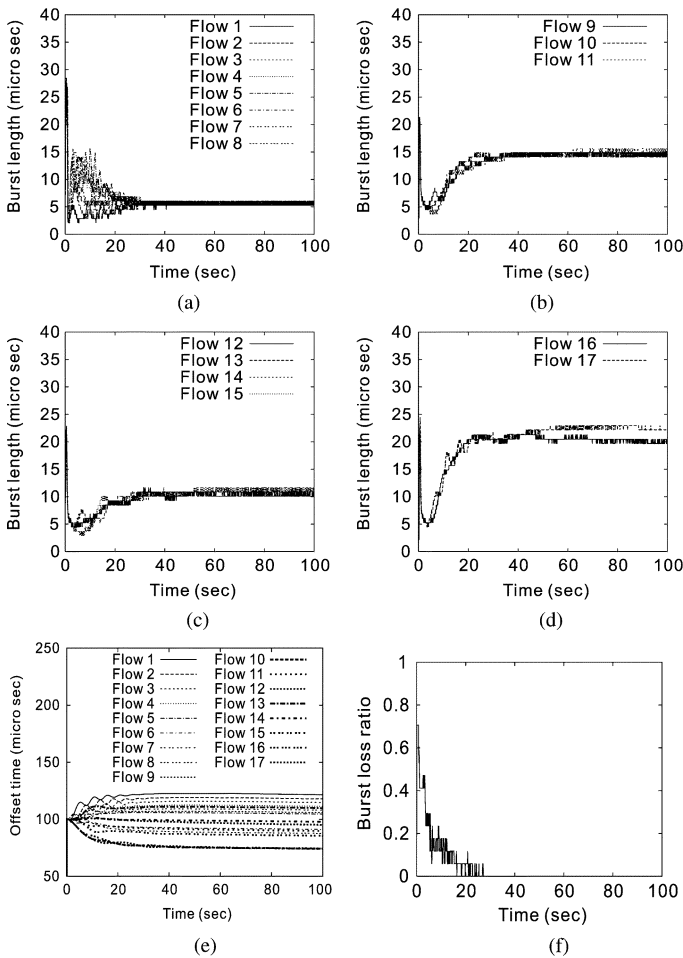
Fig. 9. Network convergence in the multiple-link scenario: Each flow group achieves different burst length. After some transient time, the network converges to the ideal state of zero loss by adjusting the burst length and offset time. (a) Burst lengths: FG1. (b) Burst lengths: FG2. (c) Burst lengths: FG3. (d) Burst lengths: FG4. (e) Offset time. (f) Burst loss ratio.
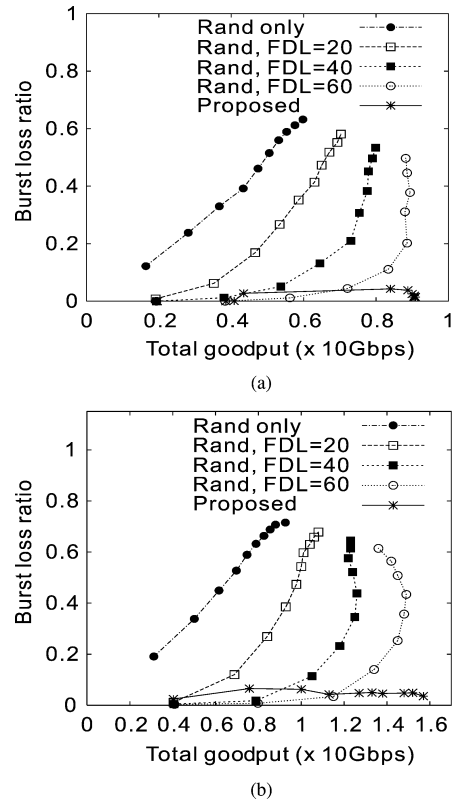

Fig. 10. Comparison of the proposed method and conventional methods while increasing offered load: In conventional methods, the goodput performance is increased only at the cost of high burst ratio. Proposed method is keeping very low burst loss ratio even at the highest goodput. (a) Single-link scenario. (b) Multiple-link scenario.

and the multiple-link scenario in Fig. 10(b). We can see that the conventional methods achieve higher goodput as the offered load increases, but unfortunately the burst loss ratio also increases. The same is true even when adopting FDL, although the performance is considerably improved as the capacity of FDL increases. However, the proposed joint control algorithm does not follow this tendency, i.e., it always achieves very low burst loss ratio always whatever the offered load is. This is because the proposed joint control algorithm does not send bursts randomly but carefully chooses the burst length and the offset time taking into account the link congestion and burst contention.

Next, we conduct the performance comparison in the NSF mesh network topology. The topology and the set of burst flows are depicted in Fig. 11(a), and we use the shortest path algorithm to select the route of each burst flow. Fig. 11(b) shows the burst loss ratio and goodput performance curves in the NSF topology when using the conventional method and the proposed method. We can see that the network topology does not affect the performance of the proposed algorithm and the conventional scheme experiences serious burst loss problems when the offered load increases.

Lastly, we compare the edge-to-edge burst delay performance using the multi-link topology in Fig. 8. For this comparison, we implemented an automatic repeat request (ARQ) algorithm such that when an ingress edge node buffer detects the burst loss (through an RCP), it can retransmit the corresponding copy of the lost burst. Fig. 12 compares the burst delays of burst flow 6. Observe that the randomized offset time control algorithm experiences severe fluctuations whereas our algorithm achieves almost constant low delay owing to zero loss in steady state. Hence, it is evident that the end-to-end delay can be significantly improved (and easily predicted) using our algorithm.

In conclusion, the proposed algorithm achieves high goodput, low burst loss rate and (almost constant) low edge-to-edge burst delay by simultaneously control the network congestion and the access time of a burst to the optical wavelength resource. Because this result is attained without expensive FDL and wavelength converter, our algorithm is superior to other schemes from not only the performance but also the cost viewpoint.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we considered the burst contention problem in the OBS network. The motivation for our work is the observation that the burst contention, which is the main bottleneck of OBS performance, is caused by not only the uncontrolled wavelength resource access trials but also the uncontrolled input
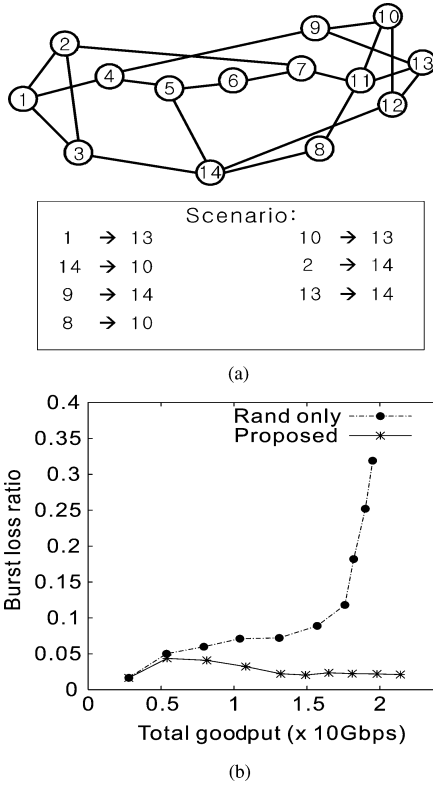
Fig. 11. Comparison of the proposed method and the conventional method at the NSF mesh topology. The proposed method outperforms in terms of burst loss ratio and goodput. (a) NSF mesh topology. (b) Performance comparison at the NSF mesh topology.
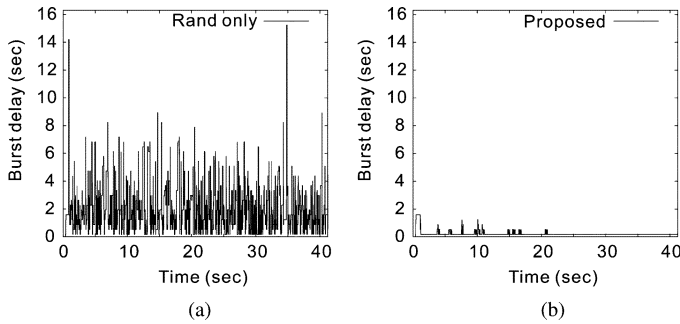

Fig. 12. Burst delay performance comparison of the burst flow 6 (in Fig. 8) between when using randomized offset time control and when using proposed algorithm in the multi-link parking lot topology. The proposed method has almost constant low delay compared with the randomized offset time control owing to no burst loss in steady state. (a) Randomized offset time only. (b) Proposed algorithm.

traffic. We proposed a joint congestion control and burst contention resolution algorithm that completely eliminates the burst contentions by explicitly controlling the network congestion and burst contention. The proposed algorithm is developed by using network utility maximization problem with optical link capacity constraints and burst contention-free constraints. Through extensive simulations, we showed that our algorithm achieves zero burst loss and high throughput at steady state, and outperforms the conventional burst contention resolution schemes.

The proposed algorithm is naturally decomposed into transport layer and medium access control (MAC) layer operations.

In the transport layer, we control the amount of incoming edge-to-edge traffic for the congestion control. At the same time, in the MAC layer, we explicitly adjust the resource access time by using the additional offset time for contention-free burst delivery. To the best of our knowledge, this is the first systematic approach to the cross-layer design of transport and MAC layer in OBS networks.

Our result shows that the performance of OBS network can be significantly improved by jointly controlling congestion and burst contention, but a number of issues have to be addressed in order for our scheme to work in practice. First, we developed edge-to-edge control, assuming that the traffic sources are given with particular model. Hence, it would be interesting to extend our framework to account for end-to-end flow control (such as TCP). In particular, adjusting TCP so as to operate under our control brings out several challenges. For example, we have to decide whether the TCP connection runs end-to-end or end-to-edge. In order to fully exploit the advantage of edge-to-edge control, it will be better to have TCP run end-to-edge. But in this case, we have to carefully decide how we let the TCP source detect the network congestion (One example would be having an adaptive queue management (AQM) in the edge buffer). If TCP runs end-to-end, parameters such as timeout should be carefully selected because it is hard to predict the queueing delay at an edge buffer which throttles the end-host traffic injected into the OBS network. Due to limited space, we could not address all these issues in this paper (instead, we emulated TCP using the Pareto traffic model), and we leave it as future work.

APPENDIX A
DERIVATION OF ALGORITHM

A. Dual Algorithm

For given point (λ, μ), let $(b(\lambda, \mu), d(\lambda, \mu))$ be the optimal solution of (6) and $f_{il}(b(\lambda, \mu), d(\lambda, \mu))$ be $t^{\text{end}}_{s^i_l l} - t^{\text{st}}_{s^{i+1}_l l}$ at point $(b(\lambda, \mu), d(\lambda, \mu))$. As we will show in the next subsection, the slack variable z is eliminated while solving (6), so we do not deal with z in the optimal solution. We denote the subgradient components of $D(\lambda, \mu)$ corresponding to λ_l and μ_{il} at point (λ, μ) by $g_l(\lambda, \mu)$ and $h_{il}(\lambda, \mu)$, respectively. Then, it is easy to show that they can be written as [18]

$$g_l(\lambda, \mu) = \tau - \sum_{s \in S(l)} b_s(\lambda, \mu) \tag{19}$$

$$h_{il}(\lambda, \mu) = -\left\{ \max\left(0, f_{il}(b(\lambda, \mu), d(\lambda, \mu)) + \frac{\mu_{il}}{\kappa}\right) - \frac{\mu_{il}}{\kappa} \right\}. \tag{20}$$

For the update of λ_l, applying subgradient projection method yields (10). Since μ_{il} is a free variable, we apply the subgradient method with step size ϵ as follows:

$$\mu_{il}(t+1) = \mu_{il}(t) - \epsilon h_{il}(\lambda(t), \mu(t)). \tag{21}$$

Letting $\epsilon = \kappa$ and substituting (20) into (21) yields

$$\mu_{il}(t+1) = \max(\mu_{il}(t) + \kappa f_{il}(b(\lambda(t), \mu(t)), d(\lambda(t), \mu(t)))) \tag{22}$$

which is equivalent to (11). It is interesting to see that the subgradient method turns into the subgradient projection method.

B. Primal Algorithm

Let $f_{il}(b,d) = t_{s_l^i l}^{\text{end}} - t_{s_l^{i+1} l}^{\text{st}}$, and rearrange $L_A(\cdot)$ as

$$
\begin{aligned}
L_A(b,d,z,\lambda,\mu) = &\sum_{s \in S} U_s(b_s) + \sum_l \lambda_l \left(\tau - \sum_{s \in S(l)} b_s \right) \\
&- \frac{1}{2}\kappa \sum_l \sum_{i=1}^{N_l} \left\{ \left(f_{il}(b,d) + z_{il}^2 + \frac{\mu_{il}}{\kappa} \right)^2 - \left(\frac{\mu_{il}}{\kappa} \right)^2 \right\}.
\end{aligned}
\tag{23}
$$

Then, $D(\lambda,\mu)$ can be rewritten as

$$
D(\lambda,\mu) = \max_{b \in B, d \geq 0} L_A(b,d,\lambda,\mu)
\tag{24}
$$

where

$$
\begin{aligned}
L_A(b,d,\lambda,\mu) = &\sum_{s \in S} U_s(b_s) + \sum_l \lambda_l \left(\tau - \sum_{s \in S(l)} b_s \right) \\
&- \frac{1}{2}\kappa \sum_l \sum_{i=1}^{N_l} \left\{ \max^2\left(0, f_{il}(b,d) + \frac{\mu_{il}}{\kappa} \right) - \left(\frac{\mu_{il}}{\kappa} \right)^2 \right\}.
\end{aligned}
\tag{25}
$$

Since $L_A(\cdot, \cdot, \lambda, \mu)$ is concave and differentiable with respect to (b,d), we can easily solve (24) using gradient projection method. Using (22), we can write the gradient components as

$$
\begin{aligned}
&\nabla_{b_s} L_A(b,d,\lambda,\mu) \\
&= U_s'(b_s) \\
&\quad - \sum_{l \in L(s)} \Big\{ \lambda_l + \max\left(0, \kappa f_{I(s,l)l}(b,d) \right. \\
&\qquad \left. + \mu_{I(s,l)l} \right) \Big\} \\
&= U_s'(b_s) - \sum_{l \in L(s)} \left\{ \lambda_l + \mu_{I(s,l)l} \right\}
\end{aligned}
\tag{26}
$$

$$
\begin{aligned}
&\nabla_{d_s} L_A(b,d,\lambda,\mu) \\
&= - \sum_{l \in L(s)} \Big\{ \max\left(0, \kappa f_{I(s,l)l}(b,d) + \mu_{I(s,l)l} \right) \\
&\qquad - \max\left(0, \kappa f_{\bar{I}(s,l)l}(b,d) \right. \\
&\qquad\qquad \left. + \mu_{\bar{I}(s,l)l} \right) \Big\} \\
&= - \sum_{l \in L(s)} \left\{ \mu_{I(s,l)l} - \mu_{\bar{I}(s,l)l} \right\}
\end{aligned}
\tag{27}
$$

There are two ways to obtain the optimal solution from the above gradients. One is to use the equation made by equating the gradient to zero and the other is to use gradient projection method, which is actually a smoothed version of the former. We used the former in (12) and the latter in (13), but any of the two methods can be selected in (12) and (13).

REFERENCES

[1] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching: A new area in optical networking research," *IEEE Network*, vol. 18, no. 3, pp. 16–23, May/Jun. 2004.

[2] F. Callegati, "Optical buffers for variable length packets," *IEEE Commun. Lett.*, vol. 4, no. 9, pp. 292–294, Sep. 2000.

[3] S. Verma, H. Chaskar, and R. Ravikanth, "Optical burst switching: A viable solution for terabit IP backbone," *IEEE Network*, vol. 14, no. 6, pp. 48–53, Nov./Dec. 2000.

[4] F. Farahmand, Q. Zhang, and J. P. Jue, "Feedback-based contention avoidance mechanism for optical burst switching networks," presented at the 3rd Int. Workshop Opt. Burst Switch., San Jose, CA, Oct. 2004.

[5] C. M. Gauger, "Dimensioning of FDL buffers for optical burst switching nodes," presented at the ONDM 2002, Torino, Italy, Feb. 2002.

[6] M. Yoo, C. Qiao, and S. Dixit, "QoS performance of optical burst switching in IP-over-WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 2062–2071, Oct. 2000.

[7] B. Ramamurthy and B. Mukherjee, "Wavelength conversion in WDM networking," *IEEE J. Sel. Areas Commun.*, vol. 8, no. 7, pp. 1061–1073, Sep. 1998.

[8] H. Li and I. Thng, "Performance analysis of a limited number of wavelength converters in an optical switching node," *IEEE Photon. Technol. Lett.*, vol. 17, no. 5, pp. 1130–1132, May 2005.

[9] W. Park, M. Shin, and S. Chong, "Performance enhancement in OBS network with flow control and edge delay method," presented at the IEEE Globecomm, San Francisco, CA, Nov. 2006.

[10] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, Apr. 1998.

[11] S. H. Low, "A duality model of TCP and active queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 525–536, Aug. 2003.

[12] S. H. Low and D. E. Lapsley, "Optimization flow control—I: Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–875, Dec. 1999.

[13] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, pp. 1969–1985, Dec. 1999.

[14] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: Utility functions, random losses and ECN marks," in *Proc. IEEE INFOCOMM*, Mar. 2000, vol. 3, pp. 1323–1332.

[15] I. P. Kaminow and T. Li, *Optical Fiber Telecommunications IV-B, Systems and Impairments*, 2nd ed. New York: Academic Press, 2002.

[16] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[17] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Programming*. Belmont, MA: Athena Scientific, 1997.

[18] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. New York: Wiley, 1993.

[19] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[20] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Heidelberg, Germany: Springer, 1985.

[21] ns-2 Network Simulator. 2000 [Online]. Available: http://www.isi.edu/nsnam/ns/

[22] K. Park and W. Willinger, *Self Similar Network Traffic and Performance Evaluation*, 1st ed. New York: Wiley, 2000.

[23] M. Nakajawa, H. Kobota, K. Suzuki, E. Yamada, and A. Sahara, "Ultrahigh-speed and long-distance TDM and WDM soliton transmission technologies," *IEEE J. Sel. Topics Quantum Electron.*, vol. 6, no. 2, pp. 363–396, Mar./Apr. 2000.

Won-Seok Park received the B.S. degree in electronics and computer engineering from Hanyang University, Seoul, Korea, in 1998, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2000 and 2006, respectively.

He is currently with the Mobile Communications R&D Center of LG Electronics Inc. His research interests are in congestion control, medium access control, optical networks, and wireless networks.

Minsu Shin received the B.S., M.S., and Ph.D. degrees from the School of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1998, 2000, and 2006, respectively.

From 2006 to 2007, he was a Visiting Postdoctoral Research Fellow at the Department of Computer Science, North Carolina State University, Raleigh. He is currently with technology planning team in SK Broadband, Seoul, Korea. His research interests are in congestion control, programmable networks, and wireless networks.
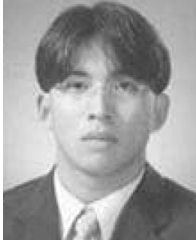
Hyang-Won Lee (S'04–A'07) received the B.S., M.S., and Ph.D. degrees, all in electrical engineering and computer science, from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2001, 2003 and 2007, respectively.

He is currently a Postdoctoral Research Associate at the Massachusetts Institute of Technology, Cambridge, MA. His research interests are in the areas of wireless networks and optical networks.

Song Chong (S'93–M'95) received the B.S. and M.S. degrees in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1988 and 1990, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin in 1995.

Since March 2000, he has been with the School of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, where he is a Professor and the Director of the Communications and Computing Group of the school. Prior to joining KAIST, he was with the Performance Analysis Department, AT&T Bell Laboratories, Murray Hill, NJ, as a Member of Technical Staff. His current research interests include wireless networks, future Internet, human mobility, and performance evaluation. He has published more than 70 papers in international journals and conferences and holds three U.S. patents in these areas.

Dr. Chong is an Editor of the *Journal of Communications and Networks* and has served on the Technical Program Committee of a number of leading international conferences, including IEEE INFOCOM, ACM CoNEXT, and ITC. He is the General Chair of WiOpt'09, and is currently the Chair of the Wireless Working Group of the Future Internet Forum and the Vice President of the Information and Communication Society of Korea.