A QoS Supporting Scheduling Algorithm for Optical Burst Switching DWDM Networks

Mei Yang and S.Q. Zheng
Department of Computer Science
Univ. of Texas at Dallas
Richardson, TX 75083-0688, USA

Dominique Verchere
Research & Innovation
Alcatel USA
Richardson, TX 75081, USA

***Abstract*-The ubiquity of IP has led to IP-over-WDM as the core architecture for the next-generation optical Internet. Optical Burst Switching(OBS) has been proposed to be a competitive switching technology for DWDM networks. Data channel scheduling Algorithm is one of the major challenges in OBS. The same-service-to-all model of the current Internet is inadequate for the diverse quality of service expectations of Internet applications and users. Differentiated Service (DiffServ) was proposed to provide a scalable and manageble architecture for service differentiation in IP networks. This paper proposes a scheduling algorithm based on an existing LAUC-VF algorithm to support DiffServ and takes advantage of MPLS. Simulation results demonstrate that this algorithm has better QoS performance than the existing LAUC-VF algorithm.**

I. Introduction

Fiber optics is a future-proof technology. In response to the exponential growth of bandwidth demand, dense wavelength division multiplexing (DWDM) has been deployed. With DWDM, it is now possible to transmit different wavelengths of light over the same fiber, which has provided another dimension to increasing bandwidth capacity. Meanwhile, the ubiquity of Internet Protocol (IP) has led to much-touted IP-over-WDM as the core architecture for the next-generation Optical Internet. Harnessing the bandwidth in an efficient and scalable way is vital to the eventual realization of such a vision.

Basically, there are three switching methods for DWDM networks: circuit switching, packet switching and burst switching. Current deployment of DWDM uses optical circuit switching (OCS), which takes the form of wavelength routing [4], [5]. Since the Internet traffic is self-similar (bursty at all time scales), wavelength routing results in a low bandwidth utilization. In addition, given the number of available wavelengths is limited, not every node can set up a dedicated lightpath to every other node. Coupled with the considerable lightpath set-up time, this limitation makes circuit switched DWDM network inflexible and unscalable for implementing IP-over-WDM. A longer-term strategy for network evolution employs optical packet switching (OPS), which provides better flexibility, resource utilization, functionality and granularity [12], [19], [20], [21], [22]. One of the biggest challenges is that there is no optical equivalence of the random access memory. At present, it seems that the most feasible way to implement optical buffering is to use fiber delay lines (FDLs). Another major challenge is the stringent requirement for synchronization, both between multiple packets, and between the header and payload of a packet. Though FDLs can be used to delay optical signals for a limited amount of time and primitive optical logic is possible, all-optical packet switching is not practically attainable for a long time. Optical burst switching (OBS) has been proposed as a combination of the merits of optical circuit switching and optical packet switching while avoiding their shortcomings. Pioneering work on optical burst switching was reported in [14], [18], and further discussions can be found in [10], [12], [13], [15], [16]. In an OBS DWDM network, the control packet (header of a burst) is transmitted on a separate wavelength ahead of the transmission of burst payload to ensure sufficient time for header processing. Compared with OCS, OBS provides improvement over wavelength routing in terms of efficiency and scalability by statistical multiplexing bursts (of packets). The pretransmission latency of OBS is lower than that of OCS. OBS requires limited or even no delay of the data at intermediate nodes as OCS, and achieves an efficient bandwidth utilization as OPS. OBS is easier to implement than OPS because of less stringent requirement in processing control signals and achieving synchronization. The amortized overhead in routing and forwarding of an OBS router can be much smaller than that of an OPS router. One of the major challenges in the design of OBS core router is the data channel scheduling algorithm. Several scheduling algorithms have been proposed to schedule bursts efficiently while achieving a high bandwidth utilization at the same time, such as LAUC-VF [6] and JET [10].

QoS is an extremely important aspect of IP-over-WDM. An IP QoS model that has been studied extensively is the integrated service (Intserv)[2]. Intserv requires a signaling mechanism, such as RSVP [3], to reserve network resources along the flow path. With Intserv, each packet is processed by the router to determine its service class. In large IP networks, processing and policing individual packets impose a computation burden on the packet forwarding engine that limits the scalability of Intserv. The differentiated services (Diffserv) model [11] was introduced to deal with the scalability issue in the Intserv model. In Diffserv, scalability is achieved by aggregating packets with the same QoS requirement into fewer but coarser-grained flows. Diffserv flows are enforced locally on per-hop basis, simplifying the complexity of end-to-end QoS policing mechanism. By closely examining the various characteristics of Multi-protocol Label Switching (MPLS) [17], one can see that it is a very good candidate for providing Diffserv [8]. Traffic classification, its ability to reserve Class of Services (CoS) through its lightweight signaling protocol LDP (Label Distribution Protocol) and label aggregation feature are some of its useful properties [7]. While MPLS was originally intro-

0-7803-7206-9/01/$17.00 © 2001 IEEE

duced as a new approach for integrating IP with ATM, it has great potential to be used for traffic engineering and Diffserv QoS in IP-over-WDM. Indeed, an extension of MPLS, Multiprotocol Lambda Switching (MPλS) was proposed recently [1] to treat crossconnects as LSRs (Label Switched Routers). Naturally, OBS/OPS DWDM routers can be extended to LSRs in OBS/OPS DWDM networks. Preliminary discussion on such possibilities can be found in [15].

In this paper, we consider QoS of IP-over-WDM. In particular, we consider how to use the Diffserv model [11] to ensure QoS on DWDM OBS networks. Lack of buffers in the DWDM layer makes existing priority schemes (Diffserv QoS model) difficult. The packet classification, burst assembly and burst scheduling at ingress routers play the most important role in QoS. In this paper, we show that, with limited buffering, data channel scheduling at core routers can improve QoS considerably. We generalize an existing scheduling algorithm to include Diffserv QoS features and take advantages of MPLS. We present simulation results of the performance of an end-to-end network. These results demonstrate that our channel scheduling algorithm at core routers significantly improve QoS performance.

II. Burst-switching DWDM Networks and LAUC-VF channel scheduling algorithm

An OBS DWDM network is a transport network that consists of edge routers and core routers. An edge router is located at the "edge" of the network, and it can be considered as a combination of an ingress edge router and an egress edge router. An ingress (resp. egress) edge router has a set of input (resp. output) links and a set of optical output (resp. input) links. Each optical link consists of a set of channels, each using a different wavelength. A core router is an intermediate node on a path between two edge routers. It has a set of optical input links and a set of output links. Each optical input/output link has a set of channels, each using a different wavelength. The channels of an optical link is divided into two groups, data channel group (DCG) and control channel group (CCG). A data channel (DC) is used to carry data bursts (DBs) and a control channel (CC) carries burst header packets (BHPs) and other control packets. IP packets received from the input of an ingress edge router are assembled into bursts based on egress edge router addresses and other information. Each data burst is transmitted along a transparent all-optical path. The transmissions of DBs and BHPs are separated. Each BHP, which carries the routing information, burst class type, offset time and burst length, is sent ahead of its DB by a nonnegative offset time. At each core router on the path leading to the destination egress router, a BHP is processed electronically to reconfigure the optical switch, allowing the DB that follows to pass without O/E and E/O conversions. While the initial value is set by the ingress edge router, the offset time may vary hop by hop as a BHP and its DB traverse across the network. The data bursts received at an egress (destination) edge router are disassembled into packets. Figure 1 shows the structure of a typical OBS core router. BHP and DB of a burst use different channels, and they go through different paths in the router. The BHP is processed by an electronic or optical switch control unit (SCU). The major functions of a SCU include BHP processing, routing table look-up to determine output links of DBs, scheduling DBs on output channels, setting up the optical switch matrix to provide input-output paths for DBs, managing limited FDL buffer (queue) to avoid transmission conflicts, switching BHPs to output control channels and regenerating BHPs.

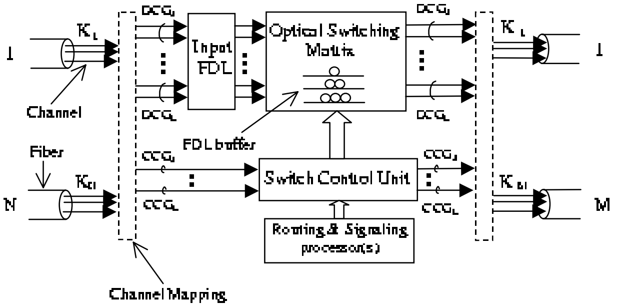

Fig. 1. A typical OBS core node.

Data channel scheduling is one of the critical functions in SCU. Without loss of generality, the following assumptions are made for channel scheduling in a core router: (1) A link has k data channels $Ch_1, Ch_2, \cdots, Ch_k$ and a control channel. This assumption simplifies our presentation. In reality, different links may have different number of channels. (2) Each output link is equipped with a scheduler, which is responsible for scheduling DBs that are switched to the link. (3) The BHPs for the DBs to be transmitted on the same output link are scheduled in the order they are received at the scheduler for that link. For simplicity, we assume that this is a linear order. (4) BHP_i includes an offset field d_i that specifies the time interval between the transmissions of BHP_i and of DB_i . (5) BHP_i includes a length field $length_i$ that specifies the length of DB_i. Assume that $length_i$ is quantified in terms of time. (6) Each DB_i is delayed by the input FDL for a fixed D time before it enters the switching matrix. This delay time is required to compensate the time for BHP processing and DB switching. (7) Recirculating FDLs support a set S_L of q available FDL delay times, $S_L = \{L_1, L_2, \cdots, L_q\}$, that can be selected by the scheduler for DBs to go through the switching matrix. (8) Slot transmission mode. In this mode, DBs are transmitted in units of slots, and BHPs are transmitted as groups, and each group is carried by one slot. Time is represented in terms of number of slots.

Let $t_s^{in}(BHP_i)$ denote the time at which the first slot of BHP_i is received, and let $t_s^{in}(DB_i)$ and $t_e^{in}(DB_i)$ denote the receiving time of the first and last slot of DB_i with respect to $t_s^{in}(BHP_i)$, respectively, as shown in Figure 2. Clearly, $t_s^{in}(DB_i) = t_s^{in}(BHP_i) + d_i$ and $t_e^{in}(DB_i) = t_s^{in}(DB_i) + length(DB_i)$.

Let $t_s^{out}(DB_i)$ and $t_e^{out}(DB_i)$ denote the transmission time of the first and last slot of DB_i over an output channel.

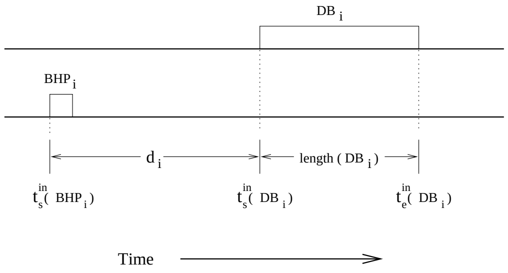

Fig. 2. Relation between the receiving times of BHP_i and DB_i.

Then, $t_s^{out}(DB_i) = t_s^{in}(BHP_i) + d_i + D + L_{r_i}$ and $t_e^{out}(DB_i) = t_s^{out}(DB_i) + length(DB_i) + D + L_{r_i}$, where L_{r_i} is an output delay time selected from $S_{L'} = S_L \cup \{L_0\}$, with $L_0 = 0$. We use $[t_s^{out}(DB_i), t_e^{out}(DB_i)]$ to denote the transmitting time interval of DB_i at a node. We say that there is a conflict between DB_i and DB_j if they are transmitted using the same output channel and $[t_s^{out}(DB_i), t_e^{out}(DB_i)] \cap [t_s^{out}(DB_j), t_e^{out}(DB_j)] \neq \emptyset$. Based on $t_s^{in}(BHP_i)$, d_i and $length(DB_i)$, $i = 1, 2, 3, \cdots$, for the same output link, the objective of channel scheduling is to assign each DB_i to an output channel Ch_u, $1 \leq u \leq k$, and select a delay time L_{r_i} from $S_{L'}$ for DB_i such that there is no conflict between DB_i and DB_j, $j \neq i$. Such a channel assignment is called a conflict-free schedule. The scheduling time (decision making time) for each (BHP_i, DB_i) pair should be no more than a bounded time. Due to the real-time constraints, limited look-ahead knowledge about future bursts, and the adopted cost/performance trade-off, it may not be always possible to find a conflict-free schedule for all DBs. Thus, we have the following alternative objective: find a conflict-free schedule of a maximum subset of $\{DB_i \mid i = 1, 2, 3, \cdots\}$ in the least possible time. The DBs that are not scheduled successfully are considered to be dropped.

Assume that BHPs arrive at the channel scheduler of an output link are BHP_i, $i = 1, 2, \cdots$, in the increasing order of i. We use set $S_I(i-1) = \{[t_s^{out}(DB_a), t_e^{out}(DB_a);\ c_a] \mid 1 \leq a \leq i-1, 1 \leq c_a \leq k\}$ to represent the state of the k channels of the link at the time DB_i is being scheduled, where $[t_s^{out}(DB_a), t_e^{out}(DB_a);\ c_a]$ represents the fact that DB_a is scheduled to be transmitted using channel Ch_{c_a} of output link with starting time $t_s^{out}(DB_a)$ and ending time $t_e^{out}(DB_a)$. Since the scheduling of DB_a, $1 \leq a \leq i-1$, is conflict-free, $[t_s^{out}(DB_x), t_e^{out}(DB_x)] \cap [t_s^{out}(DB_y), t_e^{out}(DB_y)] = \emptyset$, for $1 \leq x, y \leq i-1$ and $x \neq y$. Using $S_I(i-1)$, we define two sets, $S_M(i-1)$ and $S_G(i-1)$, which together records all time intervals that are available for future use. For $1 \leq u \leq k$, define $S_I^u(i-1) = \{[t_s^{out}(DB_a), t_e^{out}(DB_a);\ c_a] \mid [t_s^{out}(DB_a), t_e^{out}(DB_a);\ c_a] \in S_I(i-1)$ and $c_a = u\}$. Define $E_u(i-1) = \max \quad \{t_e^{out}(DB_a) \mid [t_s^{out}(DB_a), t_e^{out}(DB_a);\ c_a] \in S_I^u(i-1)\}$ if $S_I^u(i-1) \neq \emptyset$, and $E_u(i-1) = 0$ if $S_I^u(i-1) = \emptyset$.

Clearly, each $E_u(i-1)$ can be used to define a semi-unbounded time interval $[E_u(i-1), \infty)$ on channel Ch_u. We use $S_M(i-1) = \{[E_u(i-1);\ u] \mid 1 \leq u \leq k\}$ to characterize the k semi-unbounded intervals for the k channels.

A *gap* on an output channel u, represented by $g = [l, r;\ u]$, is a maximal time interval $[l, r]$ such that there is no DB transmitted on channel u with transmission time that overlaps with $[l, r]$. Clearly, a gap can be of two forms: (i) $g = [0, t_s^{out}(DB_x);\ u]$, and there does not exist any $[t_s^{out}(DB_y), t_e^{out}(DB_y);\ u]$ such that $t_s^{out}(DB_y) \leq t_s^{out}(DB_x)$, and (ii) $g = [t_e^{out}(DB_x), t_s^{out}(DB_y);\ u]$, and there does not exists $[t_s^{out}(DB_z), t_e^{out}(DB_z);\ u] \in S_I^u(i-1)$ such that either $[t_s^{out}(DB_x), t_e^{out}(DB_x)] \cap [t_s^{out}(DB_z), t_e^{out}(DB_z)] \neq \emptyset$ or $[t_s^{out}(DB_y), t_e^{out}(DB_y)] \cap [t_s^{out}(DB_z), t_e^{out}(DB_z)] \neq \emptyset$. We use $S_G(i-1)$ to denote the set of all gaps defined by schedule $S_I(i-1)$.

An algorithm, named LAUC-VF, which is the abbreviation of Latest Available Unscheduled Channel with Void Filling, the heuristics used, which was introduced in [6]. In this algorithm, a BHP reserves a wavelength for its DB based on the latest available principle, while the data burst has to be dropped if there is no available outgoing data channel after a maximum delay. A variation of LAUC-VF, which is feasible for hardware implementation was introduced in [23], is presented as follows:

LAUC-VF Algorithm

```
input: $T_i = t_s^{in}(BHP_i) + d_i + D$, $length(DB_i)$, $S_M(i-1)$, $S_G(i-1)$, and $S_L$.
output: a channel number and an output delay for $DB_(i)$, $S_M(i)$, $S_G(i)$.
begin
(a) Find a gap $[l, r;\ f]$ in $S_G(i-1)$ such that $T_i > l$ and $T_i + length(DB_i) < r$. If this operation is not successful then goto Step (b); else do the following:
    • Assign $DB_i$ to channel $Ch_f$ with $L_i = 0$;
    • $S_G(i) := S_G(i-1) \cup \{[l, T_i;\ f]\}$
    • $S_G(i) := S_G(i) \cup \{[T_i + length(DB_i), r;\ f]\}$;
    • stop.
(b) Find $[E_g;\ g]$ in $S_M(i-1)$ such that
    $T_i - E_g = \min\{T_i - E_u \mid T_i - E_u > 0, 1 \leq u \leq k\}$.
    If this operation is not successful, then go to Step (c); else do the following:
    • Assign $DB_i$ to channel $Ch_g$ with $L_{r_i} = 0$;
    • $S_M(i) := (S_M(i-1) - \{[E_g;\ g]\}) \cup \{[T_i + length(DB_i);\ g]\}$;
    • $S_G(i) := S_G(i-1) \cup \{[E_g, T_i;\ g]\}$;
    • stop.
(c) Find $[E_h;\ h]$ in $S_M(i-1)$ such that $E_h = \min\{E_u \mid 1 \leq u \leq k\}$, and find $L_a$ in $S_L$ such that $L_a = \min\{L_b \mid L_b \in S_L, L_b > E_h - T_i)\}$. If $[E_h;\ h]$ and $L_a$ are not found, goto Step (e).
(d) Find $[E_v;\ v]$ in $S_M(i-1)$ such that
    $T_i + L_a - E_v = \min\{T_i + L_a - E_u \mid E_u < T_i + L_a, 1 \leq u \leq k\}$.
(e) Find an $L_p$ in a subset $S_{L'}$ of $S_L$ and a gap $[l, r;\ w]$ in $S_G(i-1)$ such that $T_i + L_p > l$ and $T_i + length(DB_i) + L_p < r$. If this operation is not successful then goto Step (f); else do the following:
    • Assign $DB_i$ to channel $Ch_w$;
    • $S_G(i) := S_G(i-1) \cup \{[l, T_i + L_p;\ w]\}$
    • $S_G(i) := S_G(i) \cup \{[T_i + length(DB_i) + L_p, r;\ w]\}$;
    • stop.
```

```
(f) if Step (c) is not successful then drop $DB_i$ else do the following operations:
    • Assign $DB_i$ to channel $Ch_v$ with $L_{r_i} = L_a$;
    • $S_M(i) := (S_M(i-1) - \{[E_v; v]\}) \cup \{T_i + L_a + length(DB_i); v\}$;
    • $S_G(i) := S_G(i-1) \cup \{[E_v, T_i + L_a; v\}$.
    • stop
end of LAUC-VF
```

The information used by LAUC-VF scheduling algorithm is the timing of forthcoming DBs, represented by S_M and S_G. Time is an integer variable. Due to the fixed word length, relative times must be used. For fast processing, LAUC-VF is implemented by hardware using parallel processing techniques. The implementation details of LAUC-VF are discussed in [23].

III. Generalized LAUC-VF

We generalize the LAUC-VF scheduling algorithm to include Diffserv QoS features and take advantages of MPLS. We propose to partition data bursts into *n* classes depending on their QoS requirements. At an ingress edge router, a data burst is assembled by aggregating the packets of the same QoS class and with the same destination egress edge router. At a core router, data bursts going to the same output link will be scheduled by the scheduler associated with that link. For each link, its scheduler maintains *n* queues Q_1, Q_2, ... and Q_n, with Q_i being used to store the BHPs of Class-*i* DBs in the FIFO order. For each slot, the algorithm is executed once.

G-LAUC-VF Algorithm

```
begin
    for $i = 1$ to $n$ do
    while $Q_i$ has BHPs belonging to the current slot do
    begin
        $BHP_i := dequeue(Q_i)$;
        Use LAUC-VF algorithm to schedule $DB_i$
        corresponding to $BHP_i$
    end
end
```

Assuming that Class-*i* DBs have a priority higher than that of Class-j DBs if $i < j$, this algorithm ensures that for those DBs whose BHPs are in Q_i are scheduled before DBs whose BHPs are in Q_j. Note that using MPLS simpler and faster routing, forwarding and queuing can be implemented. This Generalized LAUC-VF (G-LAUC-VF) uses LAUC-VF algorithm as a subalgorithm. Simple hardware, such as a Daisy chain for requests arbitration, can be used to manage the queues. Given that LAUC-VF can be implemented in hardware, G-LAUC-VF can also be implemented in hardware. Due to the stringent real-time constraint, the simplicity of G-LAUC-VF is extremely important. Using LAUC-VF as a subalgorithm avoids the complete redesign of a new scheduling algorithm and its hardware implementation. A sophisticated queue management scheme can be easily devised, but it may be infeasible for implementation.

Source	Dest.	Routing path
0	8	0 – 3 – 4 – 5 – 8
10	9	10 – 4 – 5 – 7 – 9
1	9	1 – 3 – 6 – 7 – 9
2	8	2 – 6 – 7 – 5 – 8

TABLE I
Traffic Descriptions.

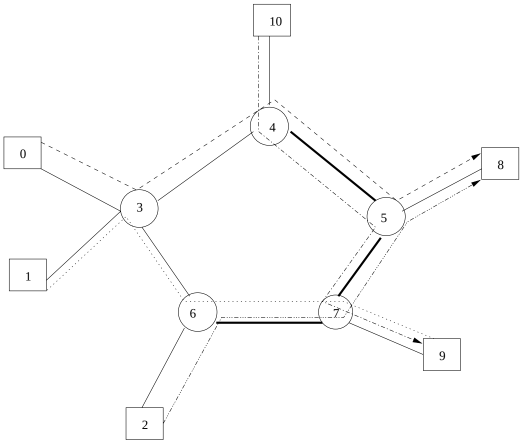

Fig. 3. An end-to-end optical network.

Now, we show the end-to-end performance of G-LAUC-VF scheduling algorithm by simulation. Considering the network topology shown in Figure 3, we assume there exist four trafficiflow groups given in Table I. Every flow group is composed of data flows of all QoS classes. Each of core nodes 5 and 7 is shared by three group of flows, and each of core nodes 3, 4, and 6 is shared by two groups of flows. Such an arrangement makes the scheduling at each core node non-trivial.

Assume that each link is composed of 8 data channels and 1 control channel, and the transmission rate on each wavelength is 1Gbps. Aslo assume that a slot takes $10\mu s$. Then, one slot can carry 1250 bytes. For simplicity, we assume that there are three classes of DBs, Class-1, Class-2 and Class-3, and their maximum data burst lengths are the same, 2 slots (i.e. 20 μs). The offset time between a BHP and its DB is a constant, 2 slots. Based on these assumptions, a control slot can certainly contain 8 BHPs, which is the maximum for a link of 8 data channels using constant burst length and constant offset time. Note that a control slot can also contain control information other than BHPs. To simulate the self-similar property of the Internet traffic, Pareto traffics [9] are generated for simulation. Data burst drops can be caused by congestion in data channels, congestion in the control channel and internal congestion in SCU. The focus of our study is on the DB drop rate due to the data channel congestion. We have generated traffic flows with

DB rate distribution of three QoS classes as (1:1:1). Data rate specified in the following is the data rate on shared paths 4-5, 5-8, 6-7 and 7-9.

Figure 4 to Figure 6 compare the drop rates of G-LAUC-VF algorithm and LAUC-VF algorithm without FDLs when the average data rate is 6Gbps and the peak rate is varying from 8Gbps to 26Gbps. While the total drop rates of these two algorithms are equal, the drop rate of Class-1 traffic is significantly reduced by the G-LAUC-VF algorithm. Table III shows the improvement percentages of the G-LAUC-VF algorithm over the original LAUC-VF algorithm for Class-1 data bursts in Figure 4. Figure 5 and Figure 6 show that the price paid for improved Class-1 drop rate is the worsened performance only for Class-3 data bursts.

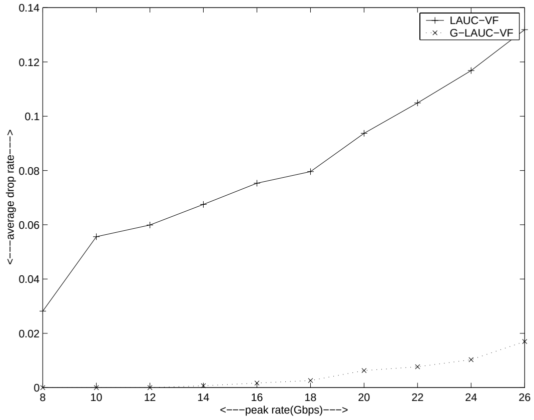

Fig. 4. Average drop rate of Class-1 DBs vs. peak data rate, assuming that the average data rate is 6Gbps.

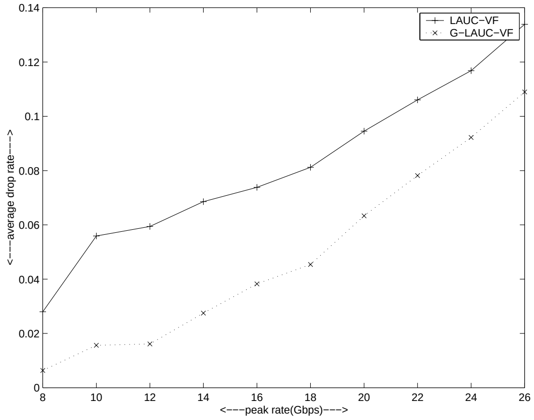

Fig. 5. Average drop rate of Class-2 DBs vs. peak data rate, assuming that the average data rate is 6Gbps.

The simulation results shown above are based on the assumption that no output buffers are used. By adding FDLs in the optical switching matrix, more flexibility is introduced to schedule DBs without conflicts. We conducted a series of simulations to look into the effects of FDLs. We assume there is a set $S_{L'}$ of q available delay slots, separated apart by one

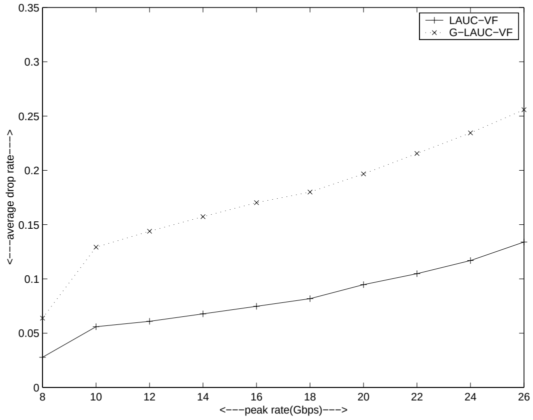

Fig. 6. Average drop rate of Class-3 DBs vs. peak data rate, assuming that the average data rate is 6Gbps.

Peak Rate (Gbps)	LAUC-VF	G-LAUC-VF	Drop Rate Improvement
8	0.0282	0.0	100%
10	0.0556	0.0	100%
12	0.0599	0.0	100%
14	0.0675	0.0007	98.96%
16	0.0753	0.0017	97.74%
18	0.0796	0.0026	96.73%
20	0.0937	0.0063	93.28%
22	0.1049	0.0077	92.66%
24	0.1168	0.0103	91.18%
26	0.1319	0.0170	87.11%

TABLE II
Drop rate improvement of G-LAUC-VF over LAUC-VF for Class-1 DBs.

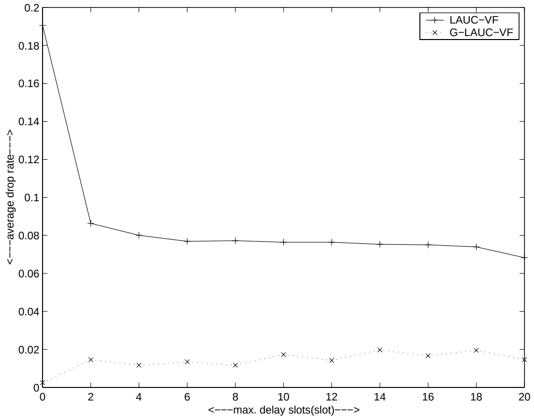

Fig. 7. Average drop rate of Class-1 DBs vs. max delay slots, assuming that the average data rate is 8Gbps and the peak data rate is 16Gbps.

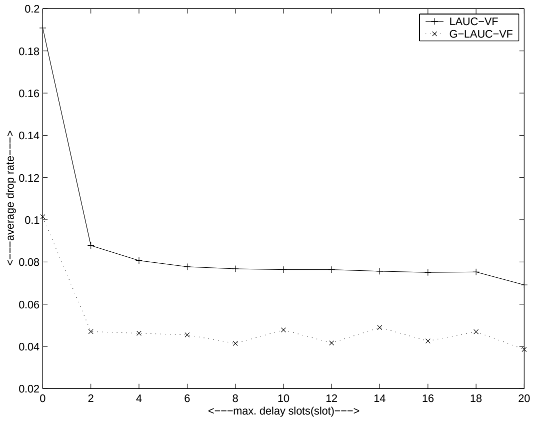

Fig. 8. Average drop rate of Class-2 DBs vs. max delay slots, assuming that the average data rate is 8Gbps and the peak data rate is 16Gbps.

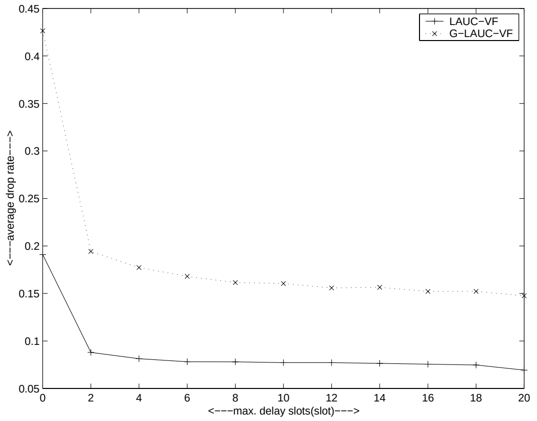

Fig. 9. Average drop rate of Class-3 DBs vs. max delay slots, assuming that the average data rate is 8Gbp and the peak data rate is 16Gbps.

slot, $S_{L'} = \{0, 1, \cdots, MaxDelaySlots\}$, where *MaxDelaySlots* is the longest delay used in the simulation. Figure 7 to Figure 9 demonstrate the drop rates of 3 classes of traffics with *MaxDelaySlots* varying from 0 slots (no FDLs) to 20 slots, when the average data rate is 8Gbps and the peak data rate is 16Gbps. These experiments indicate that, for Class-1 and Class-2 traffics the performance of G-LAUC-VF is much better than than that of LAUC-VF. However, relatively speaking, adding more output FDLs will not improve much of the drop rates of traffics with higher priorities. Since FDLs are very expensive, our results indicate that using more output FDLs is not only infeasible in practice, but also may not be desirable, as far as QoS is concerned.

IV. Concluding Remarks

In this paper, we propose a generalized LAUC-VF algorithm, which improves the QoS performance by prioritizing data bursts, maintaing multiple queues and utilizing limited optical buffers. Without loss of generality, we study the end-to-end performance of this algorithm by generating pareto traffic flows, which are accumulated into self-similiar traffic. For simplicity, we assumed that all DBs have the same length, and the offset times of DBs of all classes are the same in our simulations. We believe that it is possible the QoS performance can be further improved by assigning bursts of different classes with different lengths and different offset times. We are working on new techniques and expecting that will achieve better performance.

References

[1] D. O. Awduche, et al., “Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnect”, IETF Internet Draft, <draft-awduche-mpls-te-optical-02.txt>, Nov. 1999.

[2] R. Braden et al., “Integrated Services in the Internet Architecture”, RFC 1633, 1994.

[3] R. Braden et al., “RSVP Version 1 Functional Specification”, RFC 2205, 1997.

[4] I. Chlamtac, and G. Karmi, “Lightpath Communications: an Approach to High Bandwidth Optical WANs”, *IEEE Transactions on Communications*, vol. 40, pp. 1171-1182, July 1992.

[5] T. E. Stern, K. Bala, S. Jiang, and J. Sharony, ”Linear Lightwave Networks: Performance Issues”, *IEEE/OSA Journal of Lightwave Technology*, vol. 11, pp. 937-950, May/June, 1993

[6] Yijun Xiong, Marc Vandenhoute and Hakki C. Cankaya, “Design and analysis of optical burst-switched networks”, *SPIE Conference on All-Optical Networking: Architecture, Control and Management Issues*, Vol. 3843, 1999.

[7] F. L. Faucheur, et al. “MPLS Support of Differentiated Services”, IETF Internet Draft , <draft-ietf-mpls-diff-ext-07.txt >, Aug. 2000.

[8] Andrikopoulos, I. and Pavlou, G., “Supporting Differentiated Services in MPLS Networks”, IWQoS ’99, pp. 207-215.

[9] William Stallings, *High-Speed Networks: TCP/IP and ATM Design Principles*, Prentice Hall, 1998.

[10] M. Yoo and C. Qiao, “Just-Enugh-Time (JET): a High Speed Protocol for Burst Traffic in Optical Networks”, *Digest of IEEE/LEOS Summer Topical Meetings on Technologies for a Global Information Infrastructure*, pp. 26-27, 1997.

[11] R. Braden et al., “A Framework for Differentiated Services”, Internet draft, 1999.

[12] M. Listanti, V. Eramo, and R. Sabella, “Architectural and Technological Issues for Future Optical Internet Networks”, *IEEE Communications*, vol. 38, no. 9, pp. 82-92, 1999.

[13] S. Verma, H. Chaskar, and R. Ravikanth, “Optical Burst Switching: A Viable Solution for Terabit Backbone”, *IEEE Network*, vol. 14, no. 6, pp. 48-53, Nov.-Dec. 2000.

[14] C. Qiao and M. Yoo, “Optical burst switching(OBS)-a new paradigm for an optical Internet”, *J. High Speed Networks(JHSN)*, vol.8, no.1, pp.69-84, 1999.

[15] C. Qiao, “Labeled Optical Burst Switching for IP-over-WDM Integration”, *IEEE Communications*, vol. 38, no. 9, pp. 104-114, 2000.

[16] C.Qiao and M. Yoo, “Choices, Features and Issues in Optical Burst Switching”, *Optical Networks*, vol. 1, no. 2, pp. 36-44, 2000.

[17] E. Rosen, A. Viswananthan and R. Callon, “Multiprotocol Label Switching Architecture” < IETF Internet Draft ,draft-li-mpls-igp-arch-06.txt >, Aug. 1999.

[18] J. S. Turner, “Terabit Burst Switching”, *J. High Speed Networks*, vol. 8, no. 1, pp. 3-16, 1999.

[19] S. Yao, B. Mukherjee, and S. Dixit, “Advances in Photonic Packet Switching: An Overview”, *IEEE Communications*, vol. 38, no. 2, pp. 84-94, 2000.

[20] D. Blumenthal et al., “WDM Optical IP Tag Switching with Packet-Rate Wavelength Conversion and Subcarrier Multiplexing Addressing”, *OFC/IOOC’99. Technical Digest*, vol. 3, pp. 162-164, 1999.

[21] D. Blumenthal, P. Prucnal and J. Sauer, “Photonic Packet Switches: Architectures and Experimental Implementations”, *Proc. of the IEEE*, vol. 82, pp. 1650-1667, 1994.

[22] D. Hunter and I. Andonovic, “Approaches to Optical Internet Packet Switching”, *IEEE Communications*, vol. 38, no. 9, pp. 116-122, 1999.

[23] S.Q. Zheng et al., “Hardware Implementation of Channel Scheduling Algorithms for Optical Routers with FDL buffer”, manuscript.