Flow Control and Congestion Management for Distributed Scheduling of Burst Transmissions in Time-domain Wavelength Interleaved Networks

Andrew Brzezinski†, Iraj Saniee‡, Indra Widjaja‡, and Eytan Modiano†

†Laboratory for Information and Decision Systems, MIT
77 Massachusetts Ave, Cambridge, MA 02139
email: {brzezin, modiano}@mit.edu

‡Bell Labs, Lucent Technologies
600 Mountain Avenue, Murray Hill, NJ 07974
email: {iis, iwidjaja}@research.bell-labs.com

Abstract: This paper presents an algorithm for flow control and congestion management under the time-domain wavelength interleaved optical network architecture (described in [1]). The context of this algorithm is distributed scheduling for servicing asynchronously varying data streams.

1. Introduction

The Time-domain Wavelength Interleaved Networking (TWIN) architecture has been introduced as an efficient and cost-effective alternative to both Optical Circuit and Optical Burst Switching [1]. TWIN utilizes fast tunable lasers and burst-mode receivers at the network edge, and wavelength selective cross-connect (WSXC) for passive routing of optical signals (bursts) in the network core. A simple example of a TWIN architecture is shown in Fig. 1. Each source is equipped with a fast tunable laser and each destination is assigned a unique (set of) wavelength(s). When a source has data to send to a destination, the source tunes its laser to the wavelength assigned to that destination for the duration of the data transmission. Each intermediate node performs *self-routing* of optical bursts without buffering to the intended destination based solely on the wavelength of the burst. Self-routing is effected through use of WSXCs. No label/address lookup processing is needed in forwarding bursts from one node to another, thereby making the network core transparent and simple. The intermediate nodes are pre-configured so that any incoming optical signal of a given wavelength will be routed to the appropriate destination. One example is to pre-configure the routes that form an optical multipoint-to-point tree for each destination, as shown in Fig. 1.

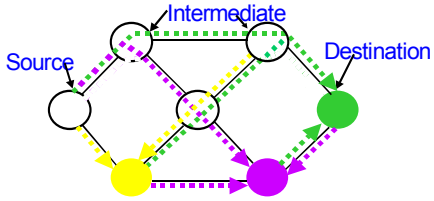

Fig. 1: TWIN architecture consisting of destination-based optical trees

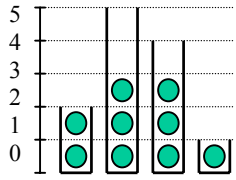

Fig. 2: Fair allocation over the backlog distribution $X_1=(X_{1,2}=2,X_{1,3}=5,X_{1,4}=4,X_{1,5}=1)$

Typically, propagation delays (100μs for ~20km) significantly dominate scheduling time-scale (~10s μs) and thus are non-negligible. Thus, although for nearly static load pre-computed centralized scheduling is feasible [2], for asynchronously varying traffic the propagation delays for a centralized scheduler may be unacceptably large. We therefore consider network control from a *distributed* scheduling standpoint, where all scheduling and flow control is performed for each node independently and on a separate control channel. The focus of this paper is on a distributed flow control algorithm for servicing asynchronous traffic in the TWIN environment.

For a network with N nodes, we consider two distributed scheduling techniques: source-based scheduling (SBS) and destination-based scheduling (DBS). Under SBS, each source node independently schedules transmissions over a pre-specified *control interval duration* of B time slots. At the (n-1)-th control timeout, source node i considers its transmission requests, and calculates vector $\mathbf{d}_i(n)=(d_{i,1}(n),d_{i,2}(n),\ldots,d_{i,N}(n))$, where $d_{i,j}(n)$ is the number of bursts to transmit from node i to node j over the next B time slots (the n-th control interval). Source i then randomly schedules these burst transmissions over the B time slots of the next control interval, never scheduling multiple bursts to be transmitted in the same time slot. Clearly, since each source performs this process independently, it is possible for multiple bursts from different sources to arrive at a particular destination in the same time slot. This *clash* of data results in the complete loss of all arrived bursts. For example, consider SBS for the case of N=3 nodes, and a control interval duration of B=8 time slots at each source, with $\mathbf{d}_1=(d_{1,2}=3,d_{1,3}=3)$ and $\mathbf{d}_2=(d_{2,1}=1,d_{2,3}=4)$. Table 1 summarizes the randomly scheduled time slots for $\mathbf{d}_1$ and $\mathbf{d}_2$ at sources 1 and 2, respectively (over only B=8 time slots of the control interval). Then, setting $\delta_{i,j}$ equal to the propagation delay between nodes i and j, with $\delta_{1,3}=1$ and $\delta_{2,3}=2$, the resulting arrivals and clashes at destination 3 are provided.

DBS is implemented similarly, with scheduling performed independently by each destination. Each source uses the control channel to convey transmission requests to each destination. Each destination independently schedules bursts over control intervals, using the control channel to return the granted schedule slots to each source. Under DBS a source can receive schedules from multiple destinations demanding multiple transmissions over a common time

interval. In this case, the source has an opportunity to resolve this *collision* by selecting a single destination to transmit to. Thus a higher throughput can be achieved under DBS, at the expense of increased scheduling delay.

Time slot	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Source 1 transmits to destination	2		3	3	2		2	3		
Source 2 transmits to destination	3	3	3			1	3			
Destination 3: burst arrives from source			2	1,2	1,2				1,2	

Table 1: Scheduled time slots under SBS and resulting arrivals for a single control interval duration.

The scheduling problem of interest is how to select $\mathbf{d}_i(n)$ for i=1,…, N over all time. Because of clashes (SBS) and conflicts (DBS), there is need for *feedback* to ensure that lost or conflicted bursts are retransmitted until receipt is acknowledged. Furthermore, since queues grow asynchronously (due to asynchronous exogenous arrivals and clashes or conflicts that arise from random independent scheduling), a *congestion* management mechanism must be incorporated into the scheduler. The XCP transport-layer protocol serves as a useful example: it makes use of link capacity, queue backlog, and transmission request information to implement an effective congestion control mechanism. Because of the bipartite nature of TWIN (bursts travel from source to destination effectively in one hop), there is no network interference: losses effectively occur at sources or destinations. Furthermore, congestion control is aided by the fact that arrival and clash/conflict information is readily collected for each source-destination (s-d) pair in the network. We present an algorithm with the following features: 1) scheduling decisions are based on recent arrival, backlog, and clash/conflict information; 2) fairness is enforced, in that no queue is allowed to be significantly starved of service; 3) explicit feedback from clashes/conflicts is used to adjust rates of service at each s-d pair.

2. System Variables and Fairness

We will restrict the following description to SBS for brevity. We refer to the virtual output queue of bursts awaiting transmission from node i to node j by $VOQ_{i,j}$. Let $A_{i,j}(n)$ be the total exogenous arrivals to $VOQ_{i,j}$ by time slot n, $C_{i,j}(n)$ be the total internal arrivals (from clashes) to $VOQ_{i,j}$ by time slot n, and $D_{i,j}(n)$ be the total transmission attempts of bursts from $VOQ_{i,j}$ made by time slot n (this includes retransmissions). Further, let $a_{i,j}(k)=A_{i,j}(kB)-A_{i,j}((k-1)B)$ be the *incremental* arrivals over the k-th control interval, and let $c_{i,j}(k)$ and $d_{i,j}(k)$ be defined similarly. Finally, let $X_{i,j}(n)$ be the total number of untransmitted bursts in $VOQ_{i,j}$ at time slot n. The scheduling problem is then to select $d_{i,j}(k+1)$ for all i,j at each time kB, k=0,1,2,….

Suppose $X_{i,j}$ is the number of bursts at source i awaiting transmission to node j. A *fair* allocation of bursts to be scheduled is achieved through *max-min fairness*. For source i, max-min fairness is accomplished by allocating the maximum number of bursts (up to the maximum B bursts of the control interval) such that for destination j either all $X_{i,j}$ bursts are allocated, or the allocation is at some maximum common level (water filling). As an example, consider Fig. 2. We wish to create a fair allocation for source 1 when N=5 nodes and B=9 time slots, and $\mathbf{X}_1=(X_{1,2}=2,X_{1,3}=5,X_{1,4}=4,X_{1,5}=1)$. For each destination j, a bowl is created with height equal to $X_{i,j}$. Then, the total of B time slots is allocated over these bowls such that each bowl is either full or at some maximum common level (in the example, the maximum level is 3). The allocation must be adjusted to be integer-valued. We denote by $d^f(\mathbf{X}_i)$ the fair allocation over the backlog vector $\mathbf{X}_i$. Thus from Fig. 2, $d^f(\mathbf{X}_1)=(2,3,3,1)$.

Unfortunately, a max-min fair allocation alone is not stable, because some backlog distributions may result in queues that cannot stop growing. To demonstrate this instability, consider N=3 nodes, B=8 time slots, initial backlogs of $X_{2,1}(0)=X_{3,1}(0)=8$ and $X_{i,j}(0)=0$ for all other i,j. Further, let there be nonzero arrival rate to $VOQ_{2,1}$ and $VOQ_{3,1}$, but no arrivals to all other VOQs for all time. Suppose that there is no propagation delay in the network, $\delta_{i,j}=0$ for all i,j At the first control timeout, the fair allocation at sources 2 and 3 schedules bursts for destination 1 across the entire control interval (see Table 2). This results in all bursts clashing and requiring retransmission. Thus, $VOQ_{2,1}$ and $VOQ_{3,1}$ never have a successful transmission and grow for all time as new arrivals occur.

Time slot	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Source 2 transmits to destination	1	1	1	1	1	1	1	1
Source 3 transmits to destination	1	1	1	1	1	1	1	1
Destination 1: burst arrives from source	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3

Table 2: Scheduled time slots from max-min fairness over a single control interval under SBS.

3. Proposed Feedback-Based Control Algorithm

The instability incurred by pure max-min fairness necessitates a flow control and congestion management mechanism that relies on persistent feedback information to adjust burst allocations for s-d pairs at each control timeout. Upon the k-th control timeout at source i, our feedback-based scheduling algorithm obtains for each destination j the value

$$y_{i,j}(k)=a_{i,j}(k)+c_{i,j}(k)-d_{i,j}^c(k),$$

where $d_{i,j}^c(k)$ is the number of transmitted bursts from $VOQ_{i,j}$ over the interval that resulted in the collision data contained in $c_{i,j}(k)$ (*i.e.* $d_{i,j}^c(k) = D_{i,j}(kB-\delta_{i,j}-\delta_{j,i})-D_{i,j}((k-1)B-\delta_{i,j}-\delta_{j,i})$). The term $y_{i,j}(k)$ is thus a local estimate of how well external and internal arrivals (from clashes) are serviced by the burst allocation. This data can also be collected over multiple control intervals. The value $d_{i,j}(k+1)$ is then obtained as follows:

Algorithm: Feedback-based burst allocation algorithm for congestion control

- Initialization: $r_{i,j}(0)=0$ for all i,j
- At the k-th control timeout (k=1,2,…): $d_{i,j}(k+1)=d_{i,j}^f(X_i(kB))+r_{i,j}(k)$, where $r_{i,j}(k)$ is obtained according to the following table, for fixed constants α,β:

Case	$r_{i,j}(k)$
$y_{i,j}(k)>0$ and $c_{i,j}(k)\geq a_{i,j}(k)$	$r_{i,j}(k-1)-\max\{\alpha, \beta y_{i,j}(k)\}$
$y_{i,j}(k)>0$ and $c_{i,j}(k)<a_{i,j}(k)$	$r_{i,j}(k-1)+\max\{\alpha, \beta y_{i,j}(k)\}$
$y_{i,j}(k)\leq 0$	$r_{i,j}(k-1)+\max\{\alpha,-\beta y_{i,j}(k)\}$

The algorithm considers three possible cases at each control interval and modifies the max-min fair burst allocation in response to these cases: Case 1 indicates that too many clashes are resulting in the inability of the scheduler to service its total exogenous and internal arrivals, and thus the number of bursts allocated should be reduced to incur fewer clashes. Case 2 indicates that the number of bursts allocated is insufficient to service the dominant exogenous arrivals, and thus should be increased to service this demand. Case 3 indicates that the burst allocation satisfies the exogenous and internal arrivals, and thus the number of bursts allocated should be increased to try to further improve the throughput on link i,j. The constant $\alpha>0$ is the minimum perturbation of $r_{i,j}$ from one control interval to the next, while $\beta>0$ is the proportionality constant relating $r_{i,j}$ to the local congestion measurement value, $y_{i,j}$. These constants are chosen to affect the responsiveness of the algorithm to the feedback (in our studies we set $\alpha=1$, $\beta=1$). Note that $d_{i,j}$ is constrained by the number of available bursts for transmission. Thus, we require $0\leq d_{i,j}(k+1)\leq X_{i,j}(kB)$.

To demonstrate the effectiveness of our feedback-based scheduler under asynchronous traffic, we provide two plots in Fig. 3. At left is a plot showing a time trace of the aggregate backlog under pure max-min fairness and under the feedback-based algorithm, when a particular destination is initially heavily loaded at each source (this is similar to the example demonstrating instability for max-min fairness), and exogenous arrivals are i.i.d. Bernoulli. Clearly the scheduler employing max-min fairness is unstable, with VOQs becoming quickly heavily loaded, while the feedback-based scheduler neatly reduces the initial backlog to a stationary regime. In the middle plot, we compare the feedback-based scheduler against a simple scheduler that allocates the next available time slot immediately upon an external or internal arrival. The plot shows histograms of the average transmission delay for bursts over all 20 VOQs (our simulation uses N=5), when all propagation delays are an equal nonzero value under independent Bernoulli exogenous arrivals with rate matrix shown at right in Fig 3. The simple scheduler experiences a much wider range of average transmission delays when compared to the feedback-based scheduler. This is an indication of the superior fairness properties of the feedback-based scheduler.

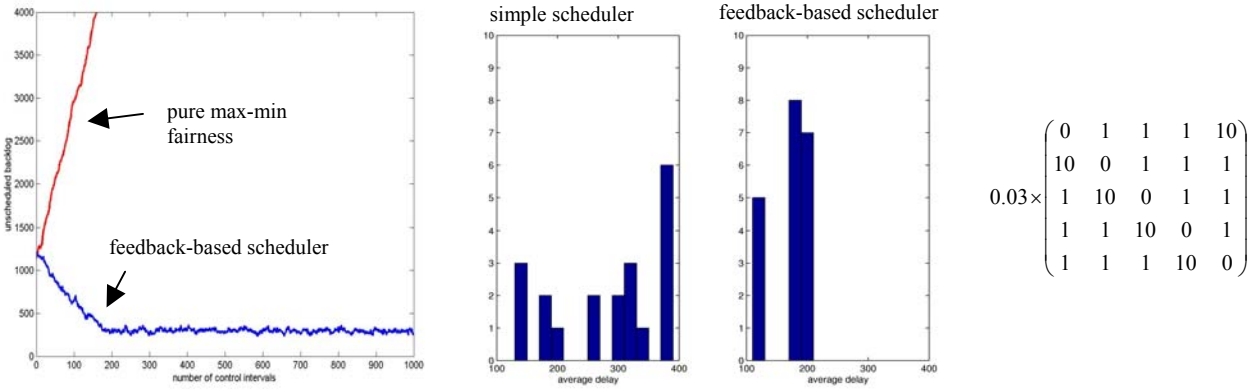

Fig. 3: Left: Time traces of two schedulers. Middle: Histograms of average delays at VOQs of two schedulers. Right: Traffic matrix for second simulation.

References

[1] I. Saniee and I. Widjaja, "A New Optical Network Architecture that Exploits Joint Time and Wavelength Interleaving," *OFC 2004*.

[2] K. Ross *et al*. "Scheduling bursts in time-domain wavelength interleaved networks," *IEEE JSAC OCN,* Vol. 21, No. 9, pp. 1441-1451, Nov. 2003.