# Flexible Bandwidth Allocation in High-Capacity Packet Switches

Aleksandra Smiljanić, *Member, IEEE*

*Abstract*— **This paper introduces a protocol for scheduling of packets in high-capacity switches, termed weighted sequential greedy scheduling (WSGS). WSGS is a simple, greedy algorithm that uses credits to reserve bandwidth for input-output pairs. By using a pipeline technique, WSGS implemented by the current technology readily supports a switching capacity exceeding 1Tb/s. Admission control is straightforward, allowing bandwidth reservations on a submillisecond time scale. Namely, the central controller readily determines if the newly requested bandwidth can be assigned to the given input-output pair. We have shown that a newly requested bandwidth should be assigned if both the input and output have enough capacity which requires checking of only two inequalities. Therefore, WSGS is well suited for switching in data networks where sessions might require high bit-rates and last for a short time. The WSGS allows bandwidth reservations with fine granularity, e.g. bandwidth can be reserved for individual web-sessions, video-streams etc.**

## I. Introduction

Most generally, packet switches transfer packets from their inputs to the specified outputs. It is important to be able to flexibly share output bandwidth among the inputs. In other words, inputs should be guaranteed to get the negotiated bandwidth even if some other inputs are overloaded. A switch with output buffers is usually a set of statistical multiplexers. Packets coming from different inputs are stored in the output buffer, and transmitted according to some scheduling policy. For example, the weighted round-robin (WRR) policy would provide to the inputs their reserved bandwidth shares. But, the capacity of a switch with output buffers is limited by the speed of the output buffer. In contrast, the capacity of a switch with input buffers is not limited similarly because packets are stored at the line bit-rate. So, switches with input buffers can provide a much higher switching capacity, which is why they have attracted much recent interest (they are discussed in most of the references [1]-[21]). In a switch with input buffers, a packet competes not only with the packets of other inputs bound for the same output, but also with the packets of the same input bound for other outputs. Several proposed protocols calculate the maximal matching between inputs and outputs that does not leave input-output pairs unmatched if there is traffic between them [1], [10], [16], [18]. However, they do not provide flexible sharing of the output bandwidth among the inputs in a switch with input buffers. Few protocols have been proposed for this purpose [1], [6], [14], [17]. We propose a new protocol which is simpler than previous proposals, and can consequently support packet switching of higher capacity. We discover that the maximal matching of inputs and outputs not only removes head-of-line (HOL) blocking [7], but also simplifies flexible bandwidth sharing in a switch with input buffers.

The simplest way to share bandwidth in a switch with input buffering is to precompute a schedule in advance based on the reservations made in a connection setup phase [1]. Time is divided into frames that consist of time slots. The schedule determines input-output pairs that will be connected in each time slot of a frame. Each input-output pair is assigned a certain number of time slots within a frame, which ensures the requested bandwidth share. It can be shown that requests can be accommodated as long as

$$\begin{aligned}&\sum_m a_{im} \le F,\\ &\sum_m a_{mj} \le F,\\ &1 \le i,j \le N, \end{aligned} \qquad (1)$$

where $a_{ij}$ is the number of time slots requested by input-output pair $(i,j)$, $F$ is the frame length, and $N$ is the number of input and output ports. As a result, the bandwidth reserved for input-output pair $(i,j)$ is $b_{ij} = B \cdot p_{ij} = B \cdot a_{ij}/F$, where $B$ is the line bit-rate. However, computing the schedule has a complexity on the order of $O(FN^2)$, and may become impracticable for fast varying traffic. For this reason, Anderson et al. propose the statistical matching algorithm to arbitrarily share the switch capacity [1]. In the statistical matching algorithm, output $j$ grants input $i$ with probability $p_{ij} = a_{ij}/F$. Each input chooses one output from which it received a grant in a specified probabilistic way. It has been shown that statistical matching uses 63% of the total switch capacity, or 72% if two iterations are performed. Stiliadis and Varma propose weighted probabilistic iterative matching (WPIM) instead of statistical matching [17]. They argue that the computing of several distribution functions within one time slot, as in statistical matching, becomes impractical in high-capacity switches. In WPIM, time is divided into frames, and input-output pair $(i,j)$ is assigned $a_{ij}$ credits within each frame. Namely, a counter

A part of this paper has been published in IEEE Conference on High Performance Switching and Routing 2000, where it won the Best Paper Award.

associated to input-output pair $(i, j)$ is set to $c_{ij} = a_{ij}$ at the beginning of a frame, and is decremented whenever this queue is served. Queues with positive counters compete for transmission with higher priority. They are rewarded according to the parallel iterative matching (PIM) algorithm. Remaining queues compete for the rest of the bandwidth, and they are again rewarded according to the PIM algorithm [1]. The performance of the WPIM protocol has been assessed only through simulations. Recently, Kam et al. proposed a scheduling algorithm for flexible bandwidth reservations in a WDMA optical network with input buffering [6]. If the number of wavelengths equals the number of users, such a WDMA network is equivalent to a switch with input buffering. Kam et al. also associate to each input-output queue a counter which is increased in each time slot by $p_{ij}$, and decreased by 1 if this queue has been served. Queues with positive counters compete for service, and they are served according to some efficient maximal weighted matching algorithm. For example, queues are considered for service in the order in which their counters decrease. Since it processes $N^2$ input-output pairs, this algorithm can also become a bottleneck in high-capacity switches. It was shown in [6] that this algorithm guarantees 50% of the switch capacity.

In this paper, we propose a new protocol, weighted sequential greedy scheduling (WSGS), that provides flexible bandwidth sharing in switches with terabit capacity. Terabit switches involve more than 100 ports, line bit-rates as high as 10Gb/s, and processing times (equal to packet transmission times) of $10 - 100$ns. Our approach is similar to the WPIM, only it is based on the sequential greedy scheduling (SGS) protocol instead of the PIM. In this way, the WSGS implementation is further simplified in a comparison to the WPIM. The PIM algorithm performs $\log_2 N + 3/4$ selections on average, in order to find maximal matching, and involves the full interconnection between input and output modules of the central controller. On the other hand, the SGS algorithm performs only one selection per time slot, and involves a central controller with simple structure. So, WSGS can potentially be used in a switch with a larger number of ports and/or higher line bit-rate, i.e. in a switch with a higher capacity. We prove that WSGS can flexibly allocate at least 50% of the total switch capacity.

## II. Optical Core of the High-Capacity Packet Switch

Different architectures for optical packet-switches have been proposed [8], [9], [11], [12], [20], [21]. Optical cross-connects capable of reconfiguration on the nanosecond time scale seem to be the best candidates for a switch core due to their simplicity [8], [9], [21]. Namely, the complexity and cost of the optical technology are very high, so the simplicity of the switch core is essential. Key fast switching optical devices that can be used in packet switches are semiconductor optical amplifiers (SOA) and rapidly tunable lasers. A research group at NEC demonstrated the operation of an optical packet switch that is based on SOAs [8], while the Lucent group built an optical packet switch which is based on fast tunable DBR lasers [21].

In the most straightforward design, a packet switch with $N$ inputs and $N$ outputs requires $N^2$ SOAs which are playing the role of gates. However, by combining WDM with space division multiplexing, the overall switch complexity measured in the number of SOAs is significantly reduced: the number of SOAs in a switch is decreased to $2N\sqrt{N}$ while $\sqrt{N}$ $\sqrt{N} \times \sqrt{N}$ waveguide grating routers (WGR) are added [8]. A 256×256 switch with 5ns switching time has been demonstrated. If the line bit-rate is 10Gbps, short packets of 64 bytes last 64ns and could be successfully switched in the proposed architecture. The total switching capacity in that case would be 256×10Gb/s=2.56Tb/s.

Alternatively, each input of a packet switch is equipped with a fast tunable laser which is connected to the outputs through a large WGR [21]. The fast tunable laser tunes to the wavelength that will be routed by the WGR to the packet designated destination. A 80×80 switch with switching time of 100ns has been demonstrated. Thus, the proposed architecture would switch only longer packets. But, the long switching time is the result of the driver design and not the laser limitation. It has been shown in [19] that the same laser can tune among wavelengths within less than 15ns. We will discuss in Section IV how the switching time influences the dynamics of the traffic that a switch can follow.

## III. Weighted Sequential Greedy Scheduling

### A. Protocol Description

The WPIM and WSGS protocols compare similarly as the PIM and SGS protocols. We will briefly review them for the sake of completeness. The PIM protocol consists of several iterations: all inputs send requests to the outputs for which they have packets to send, requested outputs send acknowledgments to their selected inputs, and selected inputs choose one output each [1]. Inputs and outputs that have not been selected in the previous iterations compete in the next iteration in the same way. It has been shown that the PIM algorithm finds a maximal matching after $\log_2 N + 3/4$ iterations on average [1]. Each iteration involves two selections, and all iterations have to be completed one after another within one packet transmission time. The planar and two-dimensional designs of the central controller that execute the PIM algorithm are shown in Figure 1 (a) or (b), respectively. Each input module (IM) sends a request to each output

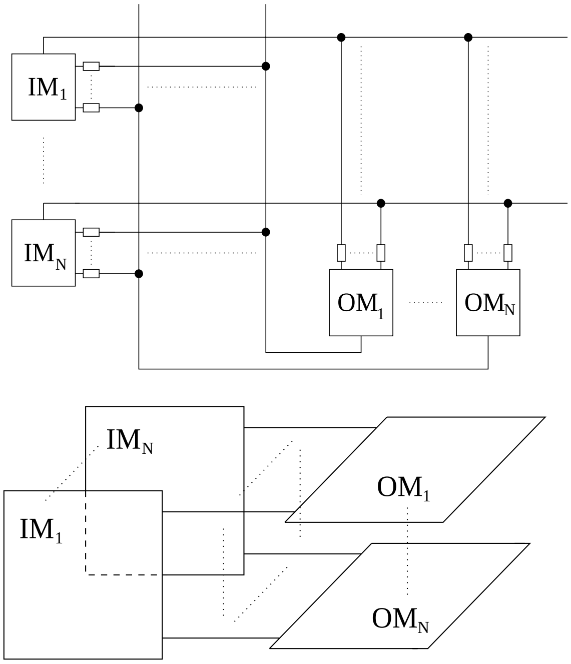

Fig. 1. Central controller for the PIM protocol

module (OM) and each OM sends an acknowledgment to each IM. There should be $2N^2$ wires connecting input and output modules. Such central controllers may become difficult for implementation as $N$ grows. On the other hand, the SGS protocol consists of $N$ steps. In the first step, some particular input chooses one of the outputs for which it has packets to send. In each following step, the next input chooses one of the remaining outputs for which it has packets to send. Clearly, SGS can be implemented by using a pipeline technique, as was discussed in [16]. For example, each step of the algorithm is completed within a separate time slot, and the algorithm is completed within $N$ time slots. Here, a time slot is the time required for packet transmission. But, in each time slot, all inputs choose outputs for different time slots in the future, so, the central controller is calculating schedules in parallel for $N$ future time slots. As a result, only one selection has to be performed within one time slot (the other $N-1$ simultaneous selections are done in parallel). In the general case of pipelining, multiple selections can be performed within one time slot, or one selection can be performed within multiple time slots. A simple structure of the central controller that executes the SGS algorithm is shown in Figure 2. Each input module communicates only with adjacent input modules, and the complex interconnection between input and output modules is avoided. Addresses of the reserved outputs are stored into the memory. The price that SGS pays for its simplicity is the additional pipeline delay, which can be as large as $N$ time slots. This pipeline delay is not critical for assumed very short packet transmission time.

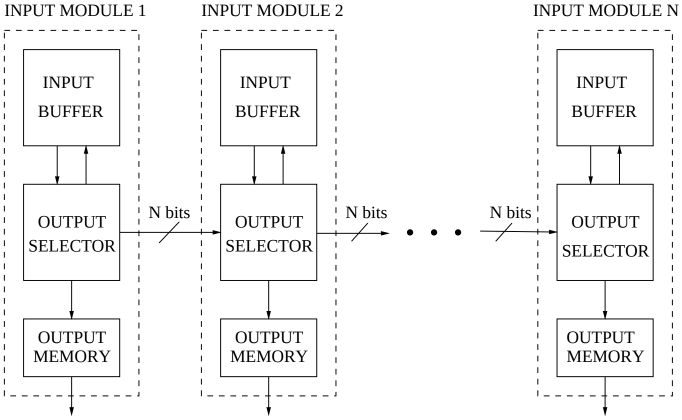

Fig. 2. Central controller for the SGS protocol

The SGS protocol needs to be further modified in order to provide flexible sharing of the total switch capacity. We propose that time is divided into frames, and the counters associated with input-output queues are set to their negotiated values at the beginning of each frame, as in WPIM. Queues with positive counters compete with higher priority according to SGS. Then, the remaining queues contend according to SGS for the available bandwidth.

Consider an $N \times N$ cross-bar switch, where each input port $i$, $i \in \{1, \cdots, N\}$, has $N$ logical queues, corresponding to each of the $N$ outputs. In each time slot, a packet can be transmitted from any chosen queue. The input of the protocol is the status of all input queues (empty/nonempty). The output of the protocol is a schedule or a set $S = \{(i,j) \mid \text{packet will be sent from input } i \text{ to output } j\}$. In any time slot, an input can only transmit one packet, and an output can receive only one packet. The schedule for the $k$th time slot is determined as follows:

- Step 1: If $k = 1 \bmod F$ then $c_{ij} = a_{ij}$;
- Step 2: $I_k = O_k = \{1, \cdots, N\}$; $i = 1$;
- Step 3: Input $i$ chooses an output, if any, $j$ from $O_k$ such that $c_{ij} > 0$, and there are unscheduled packets in the queue $(i,j)$; If there is no such $j$ go to Step 5.
- Step 4: Remove $j$ from $O_k$ and $i$ from $I_k$; Add $(i,j)$ to $S_k$; $c_{ij} = c_{ij} - 1$;
- Step 5: If $i < N$ choose $i = i + 1$ and go to Step 3;
- Step 6: $i = 1$;
- Step 7: If $i \in I_k$ choose $j$ from $O_k$ for which it has unscheduled packets to send; If there is no such $j$ go to step 9;
- Step 8: Remove $j$ from $O_k$ and $i$ from $I_k$; Add $(i,j)$ to schedule $S_k$;
- Step 9: If $i < N$ choose $i = i + 1$ and go to Step 7;

In steps 1-5, prioritized packets compete for a service according to SGS. Then, in steps 6-9, the remaining packets compete once again for the given time slot according to SGS. Steps 6-9 are optional, they will increase the efficiency of WSGS, but introduce an additional average pipeline delay of $N/2$ time slots. They represent a service for the best-effort traffic. Note that in SGS input one is always the first to pick up an output, while in the originally proposed round robin greedy scheduling (RRGS) all inputs get a chance to be the first to choose an output [16]. In the latter case an input might reserve an output in the earlier time slot for the later time slot in the future, in other words, it might interchangeably reserve outputs for different frames. So, each queue should be assigned multiple counters related to different frames.

### *B. Pipelined WSGS*

Let us first consider steps 1-5 of the pipelined WSGS. WSGS as outlined in the previous section is easy to implement by using a pipeline technique. We will assume that the output selection takes one time slot. In time slot $k$, input $i$ reserves an output for time slot $k+N+1-i$ within frame $\lceil (k+N+1-i)/F \rceil$, where $\lceil x \rceil$ is the smallest integer not smaller than $x$. Also, input $i$ resets its counters $c_{ij}$, $1 \leq j \leq N$, in time slots $lF-N+i$, where $l \geq \lceil N/F \rceil$. The time diagram for this first case of WSGS applied in a $3 \times 3$ switch is shown in Figure 3. This figure shows the relation between inputs and the time slots for which they are choosing their outputs. For example, in time slot $\mathrm{T}_4$, input $\mathrm{I}_1$ is scheduling or choosing an output for transmission during time slot $\mathrm{T}_7$, while $\mathrm{I}_3$ is scheduling for time slot $\mathrm{T}_5$ and so on. After it chooses an output, e.g., input $\mathrm{I}_1$ forwards the control information (about available outputs) to input $\mathrm{I}_2$ which reserves an output for time slot $\mathrm{T}_7$ in the next time slot $\mathrm{T}_5$. Bold vertical lines denote that input $\mathrm{I}_1$ starts a new schedule choosing any of the outputs, i.e. it does not receive the control information from input $\mathrm{I}_3$. The pipelining technique proposed in [16] that equalizes inputs might also be applied. The time diagram for this case of WSGS applied in a $3 \times 3$ switch is shown in Figure 4. Here, in each time slot another input starts a schedule. But, an input might interchangeably reserve outputs for different frames. For example, input $\mathrm{I}_1$ reserves an output for time slot $\mathrm{T}_7$ in time slot $\mathrm{T}_4$, and it reserves an output for time slot $\mathrm{T}_6$ in the next time slot $\mathrm{T}_5$. If the frame length is $F=6$ as shown in the figure, then input $\mathrm{I}_1$ interchangeably reserves outputs for frames $\mathrm{F}_2$ and $\mathrm{F}_1$. For a reasonable assumption that $F \geq N$, an input might interchangeably reserve outputs for at most two consecutive frames. So, each queue should be assigned two counters related to these two frames. Depending on the future time slot for which an input reserves an output, a specified counter of the chosen queue will be decremented by one. Counters are set every $F$ time slots.

Let us now consider all 1-9 steps of the pipelined WSGS, including service of the best-effort traffic. In any time slot $k$, each input chooses outputs for two different time slots in the future, $k+N+1-i$ and $k+2\cdot N+1-i$ within frames $\lceil (k+N+1-i)/F \rceil$ and $\lceil (k+2\cdot N+1-i)/F \rceil$. First, an input reserves an output with the positive counter for time slot $k+2\cdot N+1-i$, then, it reserves any output for time slot $k+N+1-i$. Also, input $i$ sets its counters $c_{ij}$, $1 \leq j \leq N$, in time slots $lF-2\cdot N+i$, where $l \geq \lceil 2\cdot N/F \rceil$. Figure 5 shows the time diagram for all 1-9 steps of WSGS applied in a $3 \times 3$ switch. For example, in time slot $\mathrm{T}_5$, input $\mathrm{I}_1$ chooses one of the available prioritized outputs for time slot $\mathrm{T}_{11}$, and then it chooses any of the available outputs for time slot $\mathrm{T}_8$. This is because input $\mathrm{I}_1$ uses its first chance to schedule for time slot $\mathrm{T}_{11}$ in time slot $\mathrm{T}_5$, and, therefore, it considers only queues with positive counters. On the other side, input $\mathrm{I}_1$ uses the second chance to schedule for time slot $\mathrm{T}_8$ in time slot $\mathrm{T}_5$, and, therefore, it considers all queues for service. It is possible to equalize inputs assuming service of the best-effort traffic as well.

### *C. Protocol Performance*

It is essential to determine the portion of the switch capacity that a scheduling algorithm can share among the inputs. More precisely, we want to determine the maximum admissible utilization, $p$, of any input or output line:

$$\sum_m p_{im} = \frac{1}{F}\sum_m a_{im} \leq p,$$

$$\sum_m p_{mj} = \frac{1}{F}\sum_m a_{mj} \leq p,$$

$$1 \leq i,j \leq N,$$

which can be guaranteed to the input-output pairs. So, if input-output pair $(i,j)$ requests a new portion of bandwidth, $\Delta p_{ij}$, it is accepted if:

$$\sum_m p_{im} + \Delta p_{ij} \leq p,$$

$$\sum_m p_{mj} + \Delta p_{ij} \leq p,$$

and input-output pair $(i,j)$ is assigned $\Delta a_{ij} = \lceil \Delta p_{ij} \cdot F \rceil$ new time slots per frame. We will prove that $p = 1/2$ for the WSGS, due to the fact that the SGS finds a maximal matching between inputs and outputs.

**Lemma 1:** The WSGS protocol ensures $a_{ij}$ time slots per frame to input-output pair $(i,j)$, $1 \leq i,j \leq N$, if the following condition holds:

$$\sum_m a_{im} + \sum_m a_{mj} - a_{ij} \leq F. \tag{2}$$

**Proof:** We are viewing only prioritized packets, as if WSGS consists only of steps 1-5. The reserved bandwidth

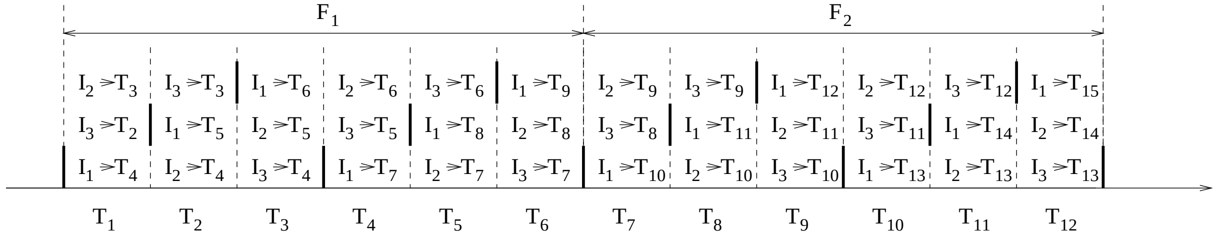

Fig. 3. Time diagram for the switch controller. $N = 3$.

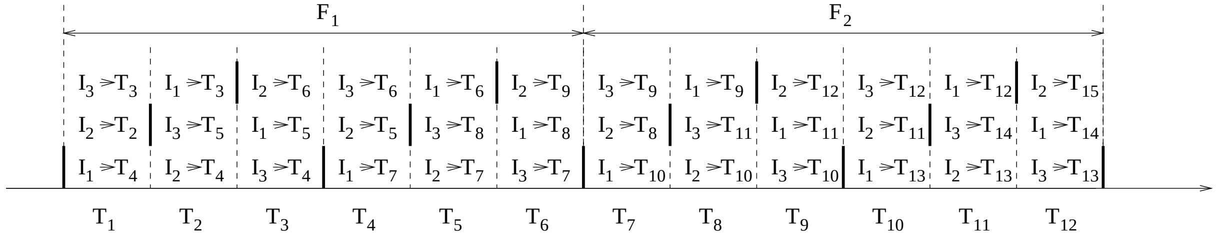

Fig. 4. Time diagram for the switch controller with input equalization. $N = 3$.

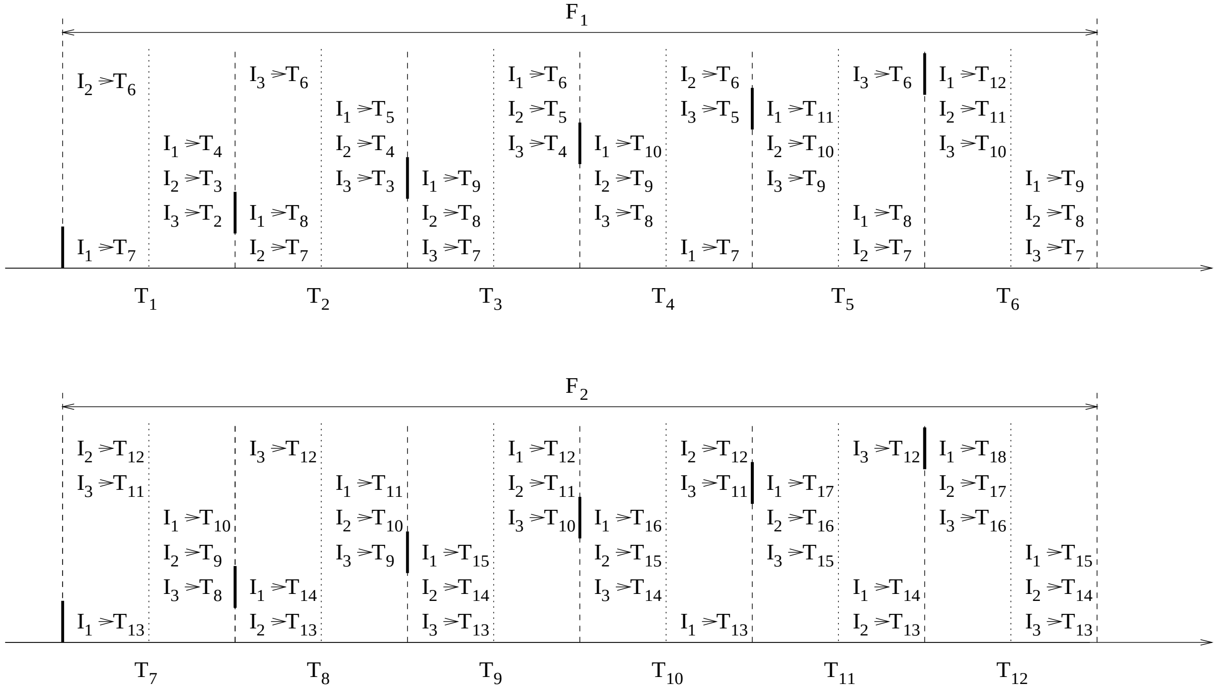

Fig. 5. Time diagram for the switch controller which serves additional best-effort traffic. $N = 3$.

is not affected by the best-effort traffic, which is served after the traffic with reservations. Observe time slots within a frame in which either input $i$ or output $j$ are connected, but not to each other. In each of these time slots, sum $s_{ij} = \sum_{m \neq j} c_{im} + \sum_{m \neq i} c_{mj}$ is greater than 0, and then it is decremented by at least 1. Sum $s_{ij}$ is the largest at the beginning of a frame and from (2), it fulfills:

$$s_{ij} = \sum_{m \neq j} a_{im} + \sum_{m \neq i} a_{mj} \leq F - a_{ij}.$$

As a conclusion, in at least $a_{ij}$ time slots per frame neither input $i$ is connected to some output other than $j$, nor output $j$ is connected to some input other than $i$. In these time slots, input $i$ reserves output $j$ if there are packets in the queue $(i, j)$ and unused credits $c_{ij} > 0$. This is because none of the inputs have chosen output $j$ before input $i$, and input $i$ is not choosing any other output. Therefore, input $i$ will choose output $j$ as supposed by SGS, and by any other algorithm that finds a maximal matching between inputs and outputs. In summary, if condition (2) is fulfilled then $a_{ij}$ time slots per frame are guaranteed to input-output pair $(i,j)$. □

**Lemma 2:** The WSGS protocol ensures $a_{ij}$ time slots per frame to input-output pair $(i,j)$, $1 \leq i,j \leq N$, if the following condition holds:

$$\begin{aligned} \sum_m a_{im} &\leq \frac{F+1}{2}, \\ \sum_m a_{mj} &\leq \frac{F+1}{2}. \end{aligned} \tag{3}$$

**Proof:** From inequality (3), it follows that:

$$\begin{aligned} \sum_m a_{im} + \sum_m a_{mj} \leq F+1 \Rightarrow \\ \sum_m a_{im} + \sum_m a_{mj} - a_{ij} \leq F, \end{aligned}$$

since $a_{ij} \geq 1$. Because inequality (3) implies inequality (2), Lemma 1 directly follows from Lemma 2. □

**Theorem:** The WSGS protocol ensures $p_{ij}$ of the line bit-rate to input-output pair $(i,j)$, $1 \leq i,j \leq N$, if the following condition holds:

$$\begin{aligned} \sum_m p_{im} &\leq \frac{1}{2}, \\ \sum_m p_{mj} &\leq \frac{1}{2}. \end{aligned} \tag{4}$$

**Proof:** Condition (4) implies (3), so Theorem follows from Lemma 2. □

Admission control in WSGS is simple, new $\Delta a_{ij}$ time slots are assigned to input-output pair $(i,j)$ if:

$$\begin{aligned} \sum_m a_{im} + \Delta a_{ij} &\leq \frac{F+1}{2}, \\ \sum_m a_{mj} + \Delta a_{ij} &\leq \frac{F+1}{2}. \end{aligned} \tag{5}$$

The central controller does not have to precompute the schedule when a new request is admitted. Only input $i$ has to update the value of $a_{ij} \leftarrow a_{ij} + \Delta a_{ij}$, $1 \leq j \leq N$, in order to set the correct counter value $c_{ij} = a_{ij}$ at the beginning of each frame. Consequently, WSGS can follow fast changes of traffic pattern.

## IV. Applications of WSGS

Let us assume that $N$ is the number of inputs and outputs, $F$ is the frame length in time slots, $B$ is the line bit-rate, and $T$ is the packet transmission time. The maximum switch throughput is:

$$C = N \cdot B. \tag{6}$$

The access time equals the frame duration:

$$A = F \cdot T. \tag{7}$$

If some input-output pair is assigned one time slot per frame, it is guaranteed the bandwidth of:

$$G = \frac{B \cdot T}{F \cdot T} = \frac{B}{F}. \tag{8}$$

So, $G$ is the granularity of bandwidth reservations. The line bit-rate $B$ and the packet transmission time $T$ are limited by the technology, and the frame length $F$ can be chosen arbitrarily. There is an apparent trade-off between the access time and the traffic granularity: by increasing $F$ the granularity is refined and the access time is prolonged and vice versa. For some realistic parameters $B$ = 10Gbps, $T$ = 50ns and chosen $F = 10^4$, the access time is $A = 10^4 \cdot 50\text{ns}=500\mu\text{s}$, and the traffic granularity is $G = 10\text{Gbps}/10^4 = 1\text{Mbps}$. So, the proposed switch can rapidly allocate bandwidth with fine granularity.

Packets generated by some source for the given destination may have to pass through multiple switches. Therefore, the bandwidth should be reserved at each of these switches. With WSGS, the bandwidth reservation at a particular switch is equivalent to the bandwidth reservation through the input and output lines in question. A switch stores the information about the bandwidth assigned to any input or output line, and would advertise this information to the other switches in the network by using interior gateway protocols (IGP) [4]. As we showed, the half of each link bit-rate can be reserved. The procedure of bandwidth reservation in wide area networks becomes very simple as suggested in [4]. Namely, links that do not have enough of the spare capacity are removed, and, then, the route is found by using the shortest path algorithm, for example. The small number of high-capacity switches would allow fast bandwidth reservation in the wide area network.

## V. Analogy between Packet and Circuit Switches

It is interesting to discuss the analogy between the bandwidth allocation in circuit and packet switches. Such an analogy was pointed out in [1]. The frame schedule in packet switches is calculated in the same way in which the connection setup is calculated in rearrangeable circuit switches. Three-stage circuit switches are built from smaller switches as shown in Figure 6. Inputs of a packet switch are equivalent to switches in the first stage of a circuit switch, outputs are equivalent to switches in the last stage, and the number of time slots assigned to some input-output pair of a packet switch are equivalent to the number of circuits between the corresponding input and output switches of a circuit switch. Finally, time slots in a frame of a packet switch correspond to switches in the

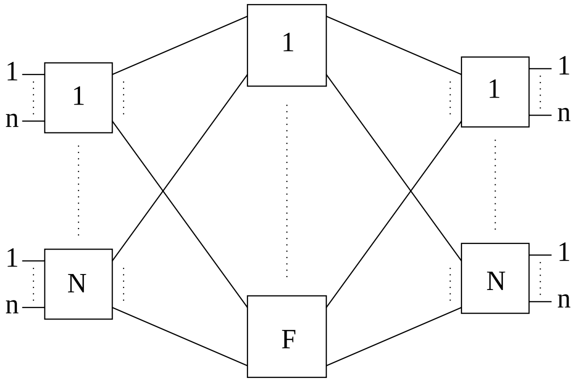

Fig. 6. Three-stage circuit switch

second stage of a circuit switch. In a circuit switch all demanded circuits should be established so that each input switch sets at most one circuit through some middle-stage switch, and also at most one circuit is set through some middle-stage switch to each output. Similarly in a packet switch, all demanded time slots should be scheduled so that each input transmits at most one packet in some time slot, and each output receives at most one packet in some time slot. The circuits in a rearrangeable three-stage circuit-switch can be set as long as:

$$n \leq F, \tag{9}$$

where $n$ is the maximum number of circuits sourced by any input switch and the maximum number of circuits destined to any output switch [5]. Condition (9) is known as the non-blocking condition in the circuit-switched theory. Considering the analogy, the time slots can be scheduled as long as an input transmits less than $F$ packets per frame, and an output receives less than $F$ packets per frame, which is given by condition (1). In other words, a new bandwidth request can be accepted if the corresponding input and output have spare capacity which exceeds the requested bandwidth amount.

However, calculating the schedule in each frame is time consuming in high-capacity switches. Therefore, we propose a greedy approach where previously assigned time slots are not rescheduled when new time slot is being scheduled. Our scheduling scheme is equivalent to the circuit setup in non-rearrangeable circuit switches. In non-rearrangeable circuit switches, circuits are also established according to the greedy algorithm. Circuits can be set if

$$2n - 1 \leq F, \tag{10}$$

which is known as a strict non-blocking condition in the circuit-switching theory [5]. Note that condition (10) is the same as condition (3) that we have derived. Also, we have derived a more general condition for admission control (2) that can be applied to circuit-switches as well.

## VI. Conclusion

We presented a very simple way to flexibly share bandwidth in switches with input buffering. The simplicity of the proposed protocol makes it attractive for switching of several Tb/s, assuming the current technology. We have also shown that the proposed WSGS can reserve 50% of the total switch capacity.

WSGS has several desirable features. First, the WSGS algorithm can serve traffic with fast varying bandwidth requirements typical in data networks. Second, WSGS requires simple processing: only two selections are to be performed within one time slot. So, it can switch short cells transmitted at high bit-rates. In addition, a linear structure of the central controller easily scales to accommodate a large number of input and output ports, and provide high switching capacity.

**A**leksandra Smiljanić (M '96) received Ph.D. and M.A. degrees in electrical engineering from Princeton University in 1999 and 1996, respectively. She completed B.Sc. in electrical engineering at Belgrade University in 1993. She has worked for AT&T Labs since 1999 on communication protocols and optical networks. She worked for two summers at NEC USA designing a packet switch with terabit capacity. Aleksandra has taught several courses at Princeton and Belgrade Universities.

Aleksandra Smiljanić is the author of the Best Paper at the IEEE Conference on High Performance Switching and Routing 2000. She is a recipient of the Aleksandar Damjanović Prize as the best student in her class at Belgrade University, 1993, and a recipient of the October Award for outstanding performance in mathematics, Belgrade 1987. Before university, she won numerous prizes in Yugoslav and international competitions in mathematics and physics.

## References

[1] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed switch scheduling for local-area networks," *ACM Transactions on Computer Systems,* vol. 11, no. 4, November 1993, pp. 319-352.

[2] H. J. Chao, "Saturn: A terabit packet switch using dual round-robin," *Proceedings of GLOBECOM*, November 2000.

[3] H. J. Chao, C. H. Lam, and X. Guo, "A fast arbitration scheme for terabit packet switches," *Proceedings of GLOBECOM*, December 1999, pp. 1236-1243.

[4] A. Ghanwani, B. Jamoussi, D. Fedyk, P.A. Smith, L. Li, N. Feldman, "Traffic engineering standards in IP network using MPLS," *IEEE Communication Magazine,* December 1999, pp. 49-53.

[5] J. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Press 1990.

[6] A. C. Kam, K. Y. Siu, R. A. Barry, and E. A. Swanson, "A cell switching WDM broadcast LAN with bandwidth guarantee and fair access," *IEEE/OSA Journal on Lightwave Technology*, vol. 16, no. 12, December 1998, pp. 2265-2280.

[7] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input vs. output queueing on a space-division packet switch," *IEEE Transactions on Communications*, vol. COM-35, no. 12, December 1987, pp. 1347-1356.

[8] Y. Maeno, Y. Suemura, A. Tajima, and N. Henmi, "A 2.56-Tb/s multiwavelength and scalable switch-fabric for fast

packet-switching networks," *IEEE Photonics Technology Letters*, vol.10, no.8, August 1998, pp. 1180-1182.

[9] Y. Maeno, Y. Suemura, S. Araki, S. Takahashi, A. Tajima, H. Takahashi, and N. Henmi, "A skew-insensitive synchronization scheme for bandwidth effective terabits per second optoelectronic packet-switch fabric," *IEEE Photonics Technology Letters*, vol.11, no. 12, December 1999, pp. 1674-1676.

[10] N. McKeown *et al.*, "The Tiny Tera: A packet switch core," *IEEE Micro*, vol. 17, no. 1, Jan.-Feb. 1997, pp. 26-33.

[11] A. Misawa, and M. Tsukada, "Broadcast-and-select photonic ATM switch with frequency multiplexed output buffers," *Journal of Lightwave Technology*, vol. 15, no. 10, pp. 1769-1777, October 1997.

[12] A. Misawa, K. Sasayama, and Y. Yamada, "WDM knockout switch with multi-output-port wavelength-channel selectors," *IEEE/OSA Journal of Lightwave Technology*, vol. 16, no. 12, pp. 2212-2219, December 1998.

[13] A. Mekkittikul, and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," *Proceedings of INFOCOM*, March 1998, pp. 792-799.

[14] R. Schoenen, and R. Hying, "Distributed cell scheduling algorithm for virtual-output-queued switches," *Proceedings of GLOBECOM*, December 1999, pp. 1211-1215.

[15] A. Smiljanić, "Flexible bandwidth allocation in terabit packet switches," *Proceedings of IEEE Conference on High Performance Switching and Routing,* June 2000, pp. 233-241.

[16] A. Smiljanić, R. Fan, and G. Ramamurthy, "RRGS-Round-Robin Greedy Scheduling for electronic/optical terabit switches," *Proceedings of GLOBECOM*, December 1999, pp. 1244-1250.

[17] D. Stiliadis, and A. Varma, "Providing bandwidth guarantees in an input-buffered crossbar switch," *Proceedings of INFOCOM*, March 1995, pp. 960-968.

[18] T. Weller, and B. Hajek, "Scheduling nonuniform traffic in a packet switching system with small propagation delay," *Proceedings of INFOCOM*, 1994, pp. 1344-1351.

[19] M. White, K. Shirkhande, M.S. Rogge, S.M. Gemelos, D. Wonglumson, G. Desa, Y. Fukashiro, and L.G. Kazovsky, "Architecture and protocol for HORNET: A novel packet-over-WDM multiple-access MAN,"*Proceedings of GLOBECOM*, November 2000.

[20] W.D. Zhong, Y. Shimazu, M. Tsukuda, and K. Yukimatsu, "A modular Tbit/s TDM-WDM photonic ATM switch using optical buffers," *IEICE Transactions on Communications*, vol. E77-B, no. 2, pp. 190-196, February 1994.

[21] J. Gripp, P. Bernasconi, C. Chan, K.L. Sherman, and M. Zirngibl, "Demonstration of a 1Tb/s optical packet switch fabric (80*12.5GB/S), scalable to 128 Tb/s (6400*20Gb/s)," *Postdeadline Paper in ECOC 2000.*