Fractional Lambda Switching™

Mario Baldi and Yoram Ofek
Synchrodyne Networks, Inc.
75 Maiden Lane, Suite 317
New York, NY10038

***Abstract* - Fractional Lambda (l) Switching (FlS™) adds the necessary efficiency to Wavelength Division Multiplexing (WDM) while preserving the simplicity of whole l switching. Due to its provisioning capability, from a fraction of STS-1 to a full optical channel capacity, FlS™ will extend the reach of optical networking all the way to the network edges in the metro and enterprise. Finally, FlS™ uniquely enables the implementation of dynamic all-optical switches since its operation does not require (1) optical processing (e.g., packet header processing) and (2) optical buffering.**

I. Introduction and Motivations

The increasing demand for communications capacity has led to the deployment of Wavelength Division Multiplexing (WDM), which requires high capacity switches. *Lambda* (λ) *switches* address this need by switching a whole wavelength from an input link to an output link without requiring any processing of the transmitted data. WDM with whole λ switching will be deployed in the network's *optical core*. However, switching of whole λs (e.g., λs of OC-192) is inefficient and costly for three reasons:

1. *N^2 problem*: the number of λs needed to accommodate all the possible connections among all access points is on the order of the square of the number of such access points. This will limit the size of the optical core, as shown in Fig. 1.
2. *Bandwidth mismatch problem*: there is a substantial *bandwidth mismatch* when extremely high capacity backbone networks feed low capacity access links. As data leave the core and are moved by packet switches towards the edge, buffers at access links frequently become congested, causing increased delays and dropped packets.
3. *Grooming and degrooming*: λ switching does not support aggregation (i.e., grooming) of traffic from multiple sources into one optical channel and separation (i.e., degrooming) of an optical channel traffic to different destinations.

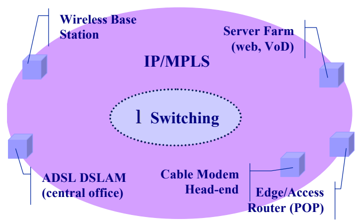

Fig. 1: Deployment of λ switching.

These three problems are solved by adding the capability of switching *fractions of l*s. This approach, which is called *Fractional l Switching* (FλS™), will permit the optical core to be extended much closer to the network edges, as shown in Fig. 2, while reaching the lower speed network access devices with a bandwidth that matches their operation capability.

FλS™ dynamically switches λ fractions while carrying data packets, in a heterogeneous, (mix of very high speed and very low speed links) meshed topology networking environment, while providing deterministic performance guarantees. λ fractions can be dynamically allocated with the proper size to satisfy the specific needs of the access networks to which a λ fraction is connected. As shown in Fig. 2, small λ fractions can be used at the periphery to access low speed sub-networks, such as, cable modems, xDSL, VoIP gateways and wireless.

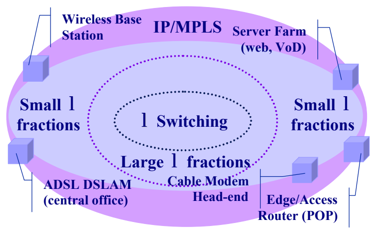

Fig. 2: Deployment of fractional λ switching (FλS).

The basic principles and the operation of FλS™ are briefly presented in Section II, while Section III analyses the complexity of fractional λ switches. The deployment of FλS for realizing dynamic optical switches is presented in Section IV, while Section V provides more rationale for FλS™.

II. The Technology: Coordinated U niversal Time (UTC) for Pipeline Forwarding

Fractional λ Switching (FλS™) combines the advantages of circuit switching and packet switching. FλS is used for constructing a Fractional λ Pipe (FλP™). A FλP™ is equivalent to a *leased line* in circuit switching. A FλP™ is realized by two simple elements:

1. Coordinated Universal Time (UTC) as a Common Time Reference (CTR™) that is globally available; and
2. Pipeline Forwarding (PF™) of time frames (logical containers of data packets) across FλPs.

A. UTC

UTC or *time of day* is the timing structure used to realize pipeline forwarding of time frames, both within switches and across FλP™s. UTC is globally available, for example through the Global Positioning System (GPS), at a low cost with an accuracy of 1 μsec. The UTC second, is partitioned into *time frames*, as shown in Fig. 3. Time frames are grouped into time cycles, and contiguous time cycles are grouped together into contiguous super cycles. The duration of a super cycle is *one UTC second*, as shown in Fig. 3, and the duration of time frames and the number of time frames in a cycle are link parameters—fast links might use shorter time frames, while slow links might use longer time frames. For example, a 1 Gb/s link might use time frames with duration of 125 μs, with time cycles of 100 time frames; while a 10 Gb/s link might use time frames with duration of 12.5 μs, with time cycles of 1000 time frames. For both links, each time frame will carry the same 15,625-byte payload, and there will be 80 time cycles in each super cycle or one UTC second.

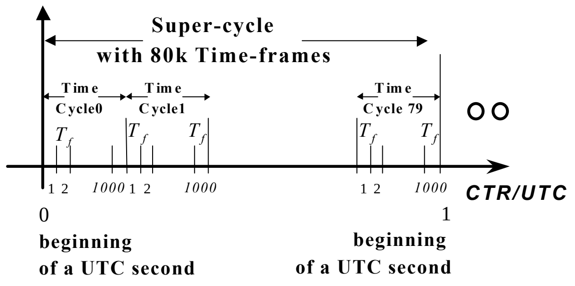

Fig. 3: UTC with TF=12.5 μs.

Time frame delimiter can be deployed in order to reduce the requirement on the UTC accuracy. For example, an idle time between the payloads of two successive time frames can be used as implicit delimiter. The UTC accuracy requirement has a direct impact on cost, stability, and implementation complexity. When a time frame delimiter are used, the UTC accuracy requirement is $½\cdot T_f$ (i.e., UTC±1/2·(10μs to 125μs)). The reason for such a relaxed requirement is that the UTC is not used for detecting the time frame boundaries, as they are detected by the idle times. Consequently, the only function of UTC is enabling is the correct mapping of the incoming time frames from the optical channel to the UTC time frames at the switch. It is easy to show that up to $½\cdot T_f$ timing error can be tolerated while maintaining the correct mapping of time frames.

Today, a time card with 1 pps (pulse per second) UTC with accuracy of 10-20 ns is available from multiple vendors. The card is small and costs $100-200. By combining such time cards with Rubidium or Cesium clocks it is possible to have a correct UTC without an external time reference for days (with Rubidium) and months (with Cesium).

B. Pipeline Forwarding – PF™

Fig. 4 shows an example of the pipeline forwarding of time frames, for a FλP™. The path through switches A, B and C has been previously scheduled and no header processing is necessary once packets enter the FλP. The time frame schedule in Switch A and B reflects a propagation delay of four time frames (time frame numbers: 2 through 5). Packets are automatically switched to the proper output port of Switch B in one time frame—reserved to the FλP in order to ensure proper operation—and then forwarded to Switch C, arriving at Switch C after three additional time frames (time frame numbers: 7 through 9). All packets are guaranteed to arrive at the end of their FλP at the same predetermined rate at which they entered the FλP.

Each FλP's switching schedule is simple, and repeats every time cycle and/or super cycle. Thus, FλP™, together with the predictability provided by UTC and pipeline forwarding, eliminate the complexity of data packet header processing. Each FλP transports data packets of one protocol, such as IP, MPLS, ATM, FR, or FC. However, FλPs on the same link may carry data packets of different protocols.

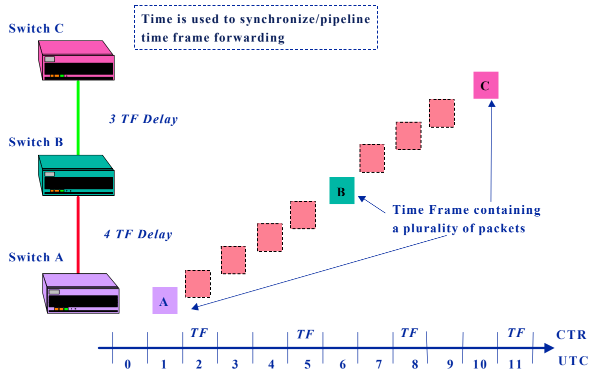
Time is used to synchronize/pipeline time frame forwarding
Switch C
Switch B
Switch A
3 TF Delay
4 TF Delay
C
B
A
Time Frame containing a plurality of packets
TF
CTR
UTC
0
1
2
3
4
5
6
7
8
9
10
11

Fig. 4: Pipeline forwarding.

III. Fractional λ Switches

Fractional λ switches have significantly lower complexity than packet switches and circuit switches with similar performances for the following reasons.

- Minimum switching fabric complexity that can be implemented using a Banyan network, which has the complexity of $a\cdot N\cdot\log_a N$ switching elements, where N is the total number of optical channels and a is the size of each switching element.
- Optimal speed-up switching fabric – the fabric operates at the same speed as the optical channel (e.g., 10 Gb/s with OC-192 links).
- Optimal memory access bandwidth equal to the optical channel bandwidth – the switch architecture enables that, with only 3 small queues, a queue is never used for reading and writing at the same time.
- (Very) small input memory for each input optical channel, e.g., a 10 Gb/s channel requires 48 Kbytes of memory, and no buffering is needed on the output port.
- (Very) simple control of the switching fabric, since its configuration changes at a relatively low frequency (e.g., 80,000 times per second) and it is known in advance. This operation complexity is comparable to

TABLE 1: MPLS, λ SWITCHING AND FRACTIONAL λ SWITCHING COMPARISON.

E.g., 16 input/output fibers, each with 16 OC-192 channels	**Packet switching (IP/MPLS) e.g., 1000 bit packets**	**Fractional l switching - FlS**	**Whole l switching**
Header processing	10 MHz	0	0
Total data units switched	$2.5\ 10^9$	$2.0\ 10^7$	0
Switching Control Speed	O(10 MHz)	80 KHz	Static
Allocation granularity	Arbitrary	Arbitrary	Whole Channel (e.g., OC-192)
Service performance	Probabilistic	Deterministic	Deterministic
Number of switching elements	At least 64K	4K (Banyan)	64K (Crossbar)

that of a T1 multiplexer.

Though highly efficient, a Banyan Network is subject to what is known as switch *blocking*: it may be impossible to connect an idle input with an idle output because a switching element is not available on the path between them. An interesting attribute of fractional λ switching is the almost complete elimination of blocking through Banyan-based switches, as shown by the following simulation results.

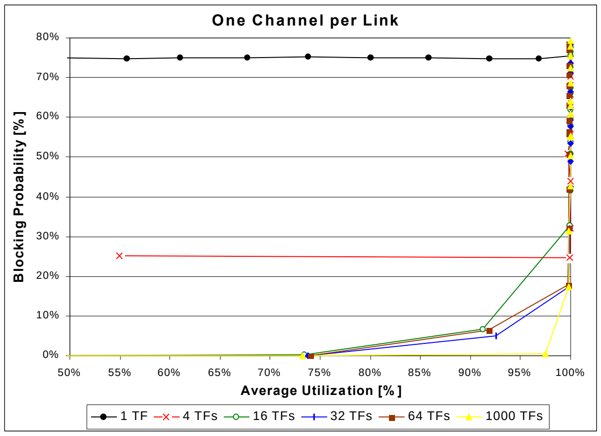

Fig. 5: Impact on blocking of number of time frames per time cycle.

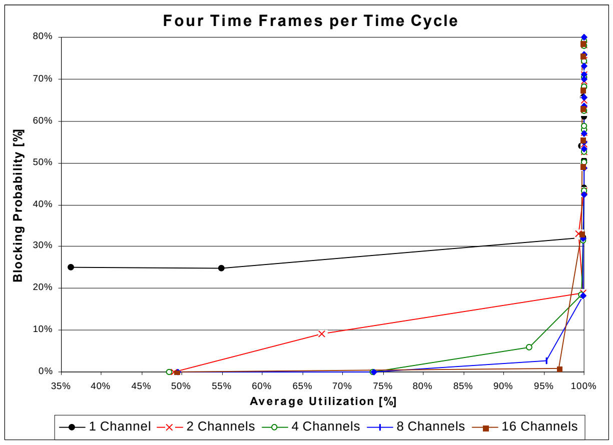

Fig. 6: Impact on blocking of number of WDM channels per optical link.

Fig. 5 plots the blocking probability of calls uniformly routed through a 4-by-4 fractional λ switch versus the average utilization of its four output links. The graph shows that with 1 time frame per time cycle, i.e., when realizing whole λ switching, the probability for a call to be blocked is 75 %. However, as the number of time frames per time cycle increases, the blocking probability decreases. With a time cycle that includes 1000 time frames calls are not rejected until the output links are loaded at over 95 %.

Deployment of multiple WDM channels on each optical link reduces blocking probability, as shown in Fig. 6 and Fig. 7. According to the results in Fig. 7, blocking is not an issue in a realistic scenario in which four WDM channels are deployed on each optical link and time cycles contain 1000 time frames. Table 1 shows the advantages of fractional λ switches over two core/backbone technologies: MPLS packet switches and whole lambda switches.

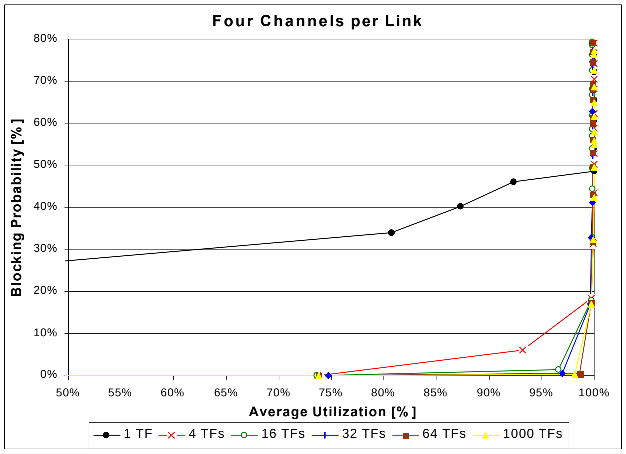

Fig. 7: Blocking probability with four channels per optical link.

Like circuit switching, e.g., SONET/SDH, FλS is using time for switching and routing. However, due to UTC and a simple alignment function (see below), fractional λ switches do not need the complex mechanisms adopted by SONET/SDH switches to cope with the drift of local clocks. Consequently, byte-by-byte multiplexing is not required on links between fractional λ switches. This enables, on one hand, packet transmission in native format within time frames and, on the other hand, deployment of explicit time frame delimiters, which enables a relaxed clock accuracy requirements (i.e., UTC$\pm\frac{1}{2}\cdot T_f$), and consequently, improved robustness in comparison with circuit switching.

Thanks to the simplicity of PF and the capability of deploying low complexity switching fabrics, electronic fractional λ switches can achieve higher performance than packet switches and circuit switches. On the other hand, the simplicity of FλS makes it very attractive for the implementation of dynamic all-optical switches.

IV. Dynamic All-Optical Switching

Dynamic all-optical switching is appealing for a number of reasons stemming from the transported data stream being transparent to the switching system: (*i*) intrinsically protocol independent (multi-protocol) transport; (*ii*) high scalability, since the transmission rate of each optical channel is transparent to the optical switching system; and (*iii*) no processing performed on switched data units, thus eliminating processing bottlenecks.

The latest advances in optical switching have resulted in decreasing reconfiguration times of optical switching fabrics. However, taking full advantage of such advances for dynamic optical switching is not obvious – for several reasons:

1. Processing of in band control information, e.g., packet headers, is not possible.
2. Dynamic optical storage is not available to assist in coping with switch control and reconfiguration time.
3. Optical switch reconfiguration time should be significantly smaller than the time between two successive reconfigurations.

Due to the above limitations it is not possible to realize an *asynchronous* switching system, and therefore, using time is necessary. However, time-based techniques deployed in circuit switching, e.g., SONET, based on byte switching and interleaving, are not applicable to all-optical switches due to the long reconfiguration time of optical switching fabrics and the lack of storage (more details in the next section).

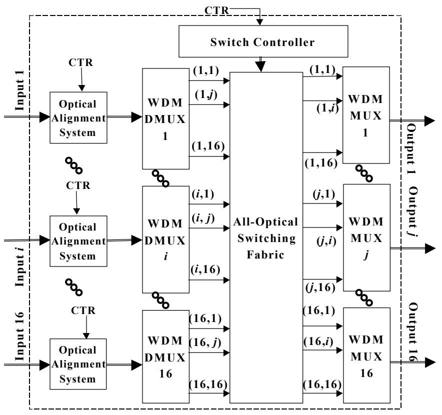

Fig. 8: All-optical Fractional λ Switch Architecture.

The most comprehensive solution to the above-mentioned problems is to use UTC for pipeline forwarding (PF™) in order to facilitate dynamic all-optical switching. UTC provides the synchronization needed to orchestrate the control of network switches while eliminating the need for processing and minimizing buffering.

Definition. *Alignment* – aligning the beginning and end of each time frame on each optical channel with the beginning and end of the UTC time frames.

Alignment is needed since the propagation delay on links between switches is not an integer multiple of time frames. The optical alignment system shown as part of the all-optical fractional λ switch in Fig. 8 operates on all the wavelengths carried by each optical fiber. The optical alignment system is based on a programmable optical delay line guaranteeing that the overall delay experienced through the optical fiber and the delay line is an integer number of time frames. As a result, when data units—that have left the switch at the transmitting end of the fiber aligned with UTC—arrive at the WDM DMUX at the receiving end they are still aligned with respect to UTC. The alignment system comprises a controller that detects time frame delimiters and adjusts the delay by using a programmable optical delay line (note that the alignment changes only when the propagation delay on the optical link changes).

V. Time and Fractional λ Switching - FλS™

This section studies the relationship between time measurements and scheduling in communications networks, in general, and FλS™, in particular. The broad approach is taken in order to provide the rationale for FλS™.

A. *Why Time?*

There are a number of answers; a simple one is that time minimizes the memory requirement, which is important for implementation in the optical domain. For example, the following comparison of the amount of silicon needed for solid state memory versus the amount of silicon needed for optical memory (i.e., optical fiber) shows that asynchronous packet switching is not practical in the optical domain.

A DRAM chip capable of storing 256 Mbits is manufactured with state of the art technology on a $10{\cdot}10^{-3}{\cdot}10{\cdot}10^{-3}{\cdot}0.5{\cdot}10^{-3}=50{\cdot}10^{-9}$ meter^3 (or $50{\cdot}10^{-6}$ Liter) silicon chip. A 256 Mbit optical memory (actually, a delay line) for an optical signal encoded at 10 Gb/s (thus, each bit is stored in $2{\cdot}10^{-2}$ meters of fiber) is realized with a $256{\cdot}10^{6}{\cdot}2{\cdot}10^{-2}=5{,}120{,}000$ meter fiber. Since the fiber diameter, core and cladding, is $125{\cdot}10^{-6}$ meter, the total volume is $\pi{\cdot}(125{\cdot}10^{-6}/2)^2{\cdot}5{,}120{,}000=62.8{\cdot}10^{-3}$ meter^3 (or 62.8 Liters).

Hence, the step from DRAM to optical memory corresponds to **an increase of more than 1,000,000 folds in the amount of silicon**. For this and other reasons extending asynchronous packet switching (which requires large buffers within network nodes) into the optical domain is not practical without some changes that reduce the memory requirements. Using time and scheduling does this.

B. *Time Measurement*

Measuring time between two events in the same location is performed locally by counting periodic rotations of various sorts. In ancient era the time was measured by counting the earth rotations, or, as some argued, the sun rotations around the earth. Since then, the measurement of time has improved dramatically.

Today, a common time reference has been established by the *time-of-day* international standard that is called *Coordinated Universal Time* or *UTC* (a.k.a. *Greenwich Mean Time* or *GMT*). Specifically, time is measured by counting the oscillations of the cesium atom in multiple locations. In fact, 9,192,631,770 oscillations of the cesium atom define one UTC second. UTC is available everywhere around the globe from several distribution systems, such as, GPS (USA satellites system), GLONASS (Russian Federation satellites system), and in the future by Galileo (European Union and Japanese satellites system).

C. Scheduling

Scheduling requires the ability to measure time. We consider scheduling with two time measurement methods:

1. Scheduling with local time based measurements. The delay between nodes cannot be measured, and therefore, the scheduling is based on local time. This method is used in circuit switching (e.g., SONET).
2. Scheduling with UTC-based measurements. The delay between nodes can be measured by using UTC and scheduling can be based on UTC. Scheduling with UTC implies no clock slips or drifts, and consequently, very simple implementation.

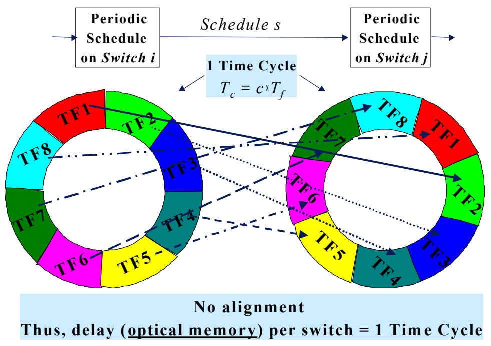
Periodic Schedule on Switch i
Schedule s
Periodic Schedule on Switch j
1 Time Cycle
$T_c = c{\times}T_f$
TF1
TF2
TF3
TF4
TF5
TF6
TF7
TF8
No alignment
Thus, delay (optical memory) per switch = 1 Time Cycle

Fig. 9: Local time based-scheduling - **SONET**

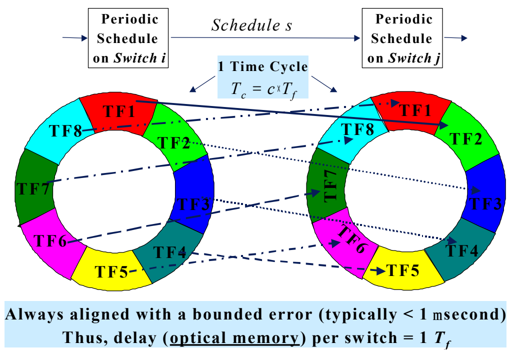
Periodic Schedule on Switch i
Schedule s
Periodic Schedule on Switch j
1 Time Cycle
$T_c = c{\times}T_f$
TF1
TF2
TF3
TF4
TF5
TF6
TF7
TF8
Always aligned with a bounded error (typically < 1 msecond)
Thus, delay (optical memory) per switch = 1 T_f

Fig. 10: UTC-based scheduling - **F1S™**

Fig. 9 and Fig. 10 are examples of the above two scheduling methods[1]. In these examples, scheduling is periodic and time is divided into time frames (TF) of predefined duration T_f. For example, a time frame of 10 µseconds is obtained by dividing one UTC second by 100,000. For periodic scheduling time frames are grouped into *time cycles*; for example, 1,000 time frames of 10 µseconds create a 10 millisecond time cycle.

D. Per Switch Delay and Optical Memory

Let's assume that two neighboring switches, *Switch i* and *Switch j*, perform a given task — e.g., switching or transmitting data units — during predefined time frames according to a schedule, *Schedule s*. *Schedule s* repeats every time cycle, T_c, where $T_c = c{\times}T_f$. In the examples in Fig. 9 and Fig. 10, $T_c = 8{\times}T_f$, and *Schedule s* on *Switch i* during time frame *k*, is scheduled on *Switch j* during time frame (*k+1*) mod *8*.

When the scheduling on *Switch i* and *Switch j* is based on local time, the delay between *Schedule s* on *Switch i* and on *Switch j* is not known, and consequently, the delay between TF *k* and TF (*k+1*) mod *c* is not known. Since the schedule repeats every time cycle, the maximum delay between a TF on *Switch i* and the corresponding TF on *Switch j* is one time cycle, T_c – where, $T_c = c{\times}T_f$.

When the scheduling on *Switch i* and *Switch j* is based on UTC, the delay between *Schedule s* on *Switch i* and on *Switch j*, is known, and consequently, the delay between TF *k* and TF (*k+1*) mod *c* is known. Consequently, the maximum time between the execution of the aforementioned task in *Switch i* and in *Switch j* is only one time frame – T_f (which results from the actual data unit propagation delay between the two switches not being an integer number of time frames – a.k.a. quantization delay). Since data units need to be stored while waiting for the task execution in *Switch j*, the time between the two task executions determines the amount of (optical) memory required within the switches.

VI. Conclusions

FλS™ uses UTC in order to implement *pipeline forwarding* (PF™) of time frames, virtual containers of 5-20 Kbytes each. Dynamic optical networking with FλS™ provides: (*i*) scalable switching with minimum complexity (i.e., Banyan network) – thereby solving the switching bottleneck, (*ii*) minimum complexity aggregation and grooming in the time domain – thereby solving the link bottleneck at the edges of the network, and (*iii*) compatibility with current public standards, such as IP/MPLS.

The efficient and deterministic bandwidth provisioning of FλS™ enables the optical core to be extended towards the edges of the network in the metro and enterprise, thus confining costly header processing to the low capacity periphery, as shown in Fig. 2. The fractional λ pipes (FλPs) realized in FλS™ networks have the same deterministic characteristics as leased lines in SONET and circuit emulation in ATM. Consequently, FλS eliminates the need for SONET — that cannot be implemented in the optical domain — or other "service" protocols, thus enabling the network "*nirvana*" in which IP/MPLS packets travel from source to destination without format conversion.

1 Without loss of generality, the propagation delay between *Switch i* and *Switch j* was ignored.