Fractional Lambda Switching:
Principles of Operation and Performance Issues

This paper introduces fractional lambda (λ) switching (FλS) and studies its blocking issues. FλS is a completely novel and extremely promising technology for the realization of low complexity, high scalability switches. FλS is unique in enabling the implementation of all-optical dynamical switching with current state-of-the-art components. Being a new technology, FλS requires investigation, especially for what its performance and efficiency are concerned, in order for its deployment to be undertaken by network operators. In this view, the present work is extremely relevant since it is instrumental in enabling a widespread deployment of such a promising technology.

Resource reservation over an FλS network requires a schedule. As in other scheduling cases, a call may not be accepted, even though there is enough capacity, because the schedule is not available—the call is then considered blocked. This work studies the probability of call blocking as a function of link utilization.

The results show that (especially if multiple wavelength division multiplexing channels are deployed on optical links between fractional λ switches) high link utilization can be achieved with negligible call blocking, even when realizing switching fabrics as Banyan networks.

This work is unique since it is related to a unique technology. Even though FλS has similarities with circuit switching (i.e., SONET/SDH switching), its basic working principle, *pipeline forwarding*, is fundamentally different since it is based on the deployment of UTC (Universal Coordinated Time) to control the forwarding of data units in the network. Pipeline forwarding provides FλS with unique features, which makes existing methodologies to study the performance of both packet switching and circuit switching not applicable to FλS. This determines the novelty and significance of this work and explains the lack of related work.

The results produced by this work can be used to encourage the deployment of FλS. Moreover, the work presents a basic approach that can be used in future FλS performance studies.

A small subset of the simulation results have been presented at the IEEE International Conference on Communications (ICC2002), Optical Networking Symposium, New York, NY, USA, Apr. 2002. However, no information on the simulation methodology and tool have been thus far published or are under revision for possible publication.

Fractional Lambda Switching
Principles of Operation and Performance Issues

Mario Baldi

Computer Engineering Department
Torino Polytechnic
Corso Duca degli Abruzzi, 24
10129 Torino – Italy
Phone: +39 011 564 7067
Fax: +39 011 564 7099
e-mail: mario.baldi @polito.it

and

Synchrodyne Networks, Inc.
e-mail: mbaldi @synchrodyne.com

Yoram Ofek

Synchrodyne Networks, Inc.
2600 Netherland Ave., Suite 1921
Riverdale, NY 10463
Phone: +1 (718) 543-1200
Fax: +1 (718) 796-8590
e-mail: ofek @synchrodyne.com

Abstract — This paper introduces fractional lambda (λ) switching (FλS) and studies its blocking issues. FλS uses a global *common time reference* (CTR) for implementing *pipeline forwarding* (PF) inside the network. A global CTR is conveniently realized with the UTC (coordinated universal time) standard. Resource reservation over an FλS network requires a schedule. As in other scheduling cases, a call may not be accepted, even though there is enough capacity, because the schedule is not available—the call is then considered blocked. This work studies the probability of call blocking as a function of link utilization. The results show that (especially if multiple wavelength division multiplexing channels are deployed on optical links between fractional λ switches) high link utilization can be achieved with negligible call blocking, even when the switching fabric is a Banyan network.

I. INTRODUCTION

The increasing demand for communications capacity has led to the deployment of Wavelength Division Multiplexing (WDM), which requires high capacity switches. *Lambda* (λ) *switches* fully address this need: they switch a whole wavelength from an input link to an output link without requiring any processing of 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. Consequently, such optical core will be relatively small.

Unlike whole λ switching, *Fractional λ Switching* (FλS) dynamically switches fractions of a λ in a heterogeneous, (mix of very high speed and very low speed links) meshed networking environment. FλS provides deterministic performance guarantees and has an implementation complexity, hence scalability, comparable to the one of whole λ switching. Very high capacity FλS switches can be implemented with off-the-shelf electronic components, as well as with all-optical switching fabrics [1][2]. Fractions of a λ, or *fractional λ pipes* (FλPs), can be dynamically allocated with the proper size to satisfy the specific needs of the access networks to which a λ fraction is connected. The FλP is equivalent to a leased line in circuit switching and can transport either packets, e.g., IP, MPLS, frame relay, ATM, Ethernet, or even SONET/SDH frames (i.e., one or more SONET/SDM channels can be mapped on an FλP).

FλS uses a global *common time reference* (CTR) for implementing *pipeline forwarding* (PF) inside the network, as explained in Section II. A global CTR is conveniently realized with the UTC[1] (coordinated

---

[1] UTC (coordinated universal time, a.k.a. Greenwich mean time or GMT) second is defined by counting 9,192,631,770 oscillations of the cesium atom. 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). There are other means for distribution of UTC, such as, radio stations, CDMA cellular telephone system and TWTFT (Two-Way Satellite Time and Frequency Transfer) technique based on communications satellites.

universal time) standard. UTC provides **phase synchronization** with **identical frequencies** everywhere, while traditional TDM (time division multiplexing) systems are using only frequency (or clock) synchronization with some clock drifts – i.e., the clock frequencies in TDM are not identical. The basic switching and forwarding units are *time frames* (TFs). TFs are generated by dividing the UTC second by a predefined set of numbers. Thus, TFs may have different durations (typically, between 5 μs and 125 μs) in different parts of the network. Section II provides a description of FλS and its basic principle of operation and comparison with other related methods.

An FλP is created by reserving switching and transmission capacity along its path during a set of TFs, which reoccurs with a predefined periodicity. This requires a feasible schedule for the time frames during which capacity is reserved on a sequence of links along a path. If there is no feasible schedule while there is available capacity, the FλP is considered blocked.

It will be shown that blocking in FλS is not significant, even when a blocking switching fabric, such as a Banyan network, is deployed. In order to study the *blocking probability* in FλS networks we developed a simulator that is described in detail in Section III. Section IV presents selected results obtained from simulations aimed at measuring link utilization and the corresponding blocking probability in a variety of scenarios. Conclusions are drawn in Section V.

A. Related Works

As already mentioned, CTR (which can be realized with UTC) provides **phase synchronization** with **identical frequencies** everywhere, while traditional TDM (time division multiplexing) systems, such as SONET/SDH, are using only frequency (or clock) synchronization with known bounds on frequency drifts. Early results on how CTR is used in packet switching were published in [1]-[7].

In order to overcome possible data loss due to the deployment of sole frequency synchronization, SONET/SDH is using a rather complex mechanism based on overhead information to accommodate: (1) the

accumulation of various delay uncertainties or jitter and (2) continuous clock drifts from a nominal value. As a consequence of sole frequency synchronization, the SONET/SDH solution implements a sophisticated forwarding of small 1-byte data units that are transported in time slots (TS) orderly organized in periodically reoccurring frames. In fact, due to lack of phase synchronization, a received data unit may have to be stored for a whole frame waiting for its TS on its outgoing link. In order to limit the latency through each switch, the frame duration has to be kept short, hence the need for small data units. For example, the TS duration in 10 Gbps is about 1 nanosecond. Given such a small data unit size, it may not be possible to realize the sophisticated SONET TDM operation in the optical domain.

One possible solution in order to make optical implementation feasible would be to increase by a factor of, say, 1000 the TS duration. However, this would increase by a factor of 1000 also the frame duration, hence the delay per switch, the accumulated jitter and optical memory requirements. Note that increasing the TS duration by a factor of 1000 will not anyway eliminate the need for processing and updating, in the optical domain, the SONET overhead information, such as, time multiplexing pointers. These pointers are needed since local time measurements on different switches are contiguously drifting from one another and jitter is being accumulated. The CTR-based switching solution presented in this paper is suitable to optical implementation since it deploys large TSs (called time frames) and does not require the sophisticated processing of overhead information characterizing SONET.

In spite of the major challenges for implementing SONET TDM in the optical domain, in the past ten years there where number of works on combining WDM with TDM [8]-[15]. None of these works was using CTR (as described in [6] and [7]) with pipeline forwarding and they do not address critical timing issues. Specifically, they do not address the accumulation of delay uncertainties or jitter and clock drifts, which is solved by either SONET with its compensation mechanisms or CTR with pipeline forwarding, as discussed in Section II.

In [8], an optical *time slot interchange* (TSI) utilizing sophisticated optical delay lines is described with no detailed timing analysis. In [9] and [10] two experimental optical system with in-band master clock distribution and optical delay lines are described, with only limited discussion of timing issues. In [11] a system with constant delays and clocks is described, which can be viewed as a close model of the *immediate forwarding*

operation mode of fractional λ switching (see Section II). However, no timing analysis and no consideration of the *non-immediate forwarding* operation mode (see Section II), that was analyzed in [3][4], are provided.

In [12] there is a TSI system with a detailed design of an optical slot permuter that does not use CTR. The system has higher implementation complexity than one realizing pipeline forwarding with CTR. The blocking probability analysis in [13] and [14] models a network with TSI switches while ignoring timing issues, such as, delays and timing errors. Thus, it is not possible to determine whether the analysis is applicable to network with CTR with *non-immediate forwarding*. The analysis in [15] provides more details than in [13] and [14] regarding timing issues in the network model, specifically, a synchronizer that aligns incoming time slots while ensuring that the delay between nodes is an integer number of time slots. The issue in this case, is how the alignment is performed, e.g., how to align when one incoming time slot starts exactly in the middle of another incoming time slot and how the synchronizer "decides" what time reference to use. Furthermore, if there are accumulated timing errors and clock drifts the synchronizer operation is complicated and not likely to be performed in the optical domain (see, for example, the timing analysis in [16]). However, if the network model and analysis in [15] is implicitly assuming both common time reference (CTR) and pipeline forwarding, then the alignment operation with respect to CTR is simple. Consequently, the analysis may be applicable to networks with CTR and pipeline forwarding as studied in this manuscript.

II.   FRACTIONAL LAMBDA SWITCHING

Fractional λ switching (FλS) consists of two basic elements that facilitate its deterministic service characteristics: (*i*) UTC that is a globally available common time reference (CTR) source; CTR is a necessary condition for (*ii*) pipeline forwarding – PF (a known optimal method that is independent of a specific realization, and is extensively used in manufacturing and computing).

A.   UTC

The UTC second is partitioned into time frames – TFs, with durations between $T_f=125/16=7.8125$ μs and $T_f=125$ μs. TFs are the basis for scheduling data unit forwarding throughout the network, as described in Section B.

Time is organized as follows: k contiguous TFs of duration T_f are grouped into a *time cycle* and l contiguous time cycles are grouped together into a *super cycle*. The TFs in a cycle are numbered from 0 to k-1 and all arithmetic expressions involving TF numbers are

modulo k; for example, if i is a TF number, then $(i+1)$ means $(i+1)$ mod k. A typical duration of a super cycle is equal to one UTC second, as shown in Figure 1 (for T_f=12.5 µs, with $k = 1000$ and $l = 80$).

A 1 pps (pulse per second) signal aligned to UTC with a 10-20 ns accuracy can be obtained from the GPS at a low cost ($100-200). However, since time frames have explicit boundaries the actual UTC accuracy requirement is: $\pm\frac{1}{2}\cdot T_f$. The reason for such a relaxed timing requirement is that UTC is not used for detecting the time frame boundaries. Consequently, the only problem is the correct mapping of the incoming TFs to the TFs that are directly derived from UTC.

B. Periodic Pipeline Forwarding

In pipeline forwarding (PF) data units are forwarded through FλS networks one hop every predefined integer number of TFs, as shown in Figure 2. Since the delay between switches is not an integer number of time frames, a UTC alignment subsystem is used to round up the delay to an integer number of TFs.

The delay experienced by data units of a given connection is predefined in a deterministic manner by imposing that the delay between an input port of one node and the input port of the next node is a predefined integer of TFs. Note that this delay includes the propagation and switching time. The maximum variation of the delay, usually called *jitter*, experienced by data units through a FλS network is one T_f.

Minimum delay is achieved by implementing *immediate forwarding*: data units of a given connection c received in TF(t) are moved to their output port and sent out in TF(t+1). However, *non-immediate forwarding* is also possible: data units of a given connection c received in TF i are moved to their output port and sent out in TF($t+k_c$) ($k_c > 1$). The flexible selection of k_c enables flexible scheduling with low blocking probability. Note that if k_c is equal to k – the *time cycle* there is no blocking. Since with $k_c \leq k$ it is always possible to find a schedule at the expense of longer delay, while preserving constant jitter of one T_f.

Note that periodic pipeline forwarding is convenient but not necessary. With UTC (time-of-day) connections may have non-periodic schedules as well. However, this will not be used in the context of this paper.

C. Switching and Alignment

Simpler switching control and higher switching fabric utilization that is independent of data unit format and switching technology is obtained by having *aligned* fixed size switching units—where transfer to the output ports of switching units starts and ends concurrently from all the input ports. TFs, the switching unit used in

FλS, have a fixed size and, as explained in the foregoing, TFs on all the switches' output links are *aligned* to UTC.

However, TFs at the input ports of a fractional λ switch are usually not aligned: TFs are aligned to UTC at the transmitting end of all links, but the propagation delay across those links will most likely not be an integer multiple of the TF duration. Hence, TFs at the input ports are aligned to the unique time reference (UTR) of the link. As shown in the fractional λ switch architecture depicted in Figure 3, data units received on the input links need to be aligned to the CTR before being transferred through the switching fabric. During each TF, a TF's worth of data units is switched from each input port to the respective output port, the transfer starting and ending concurrently for all the input/output port pairs.

Figure 3 shows a possible architecture for a fractional λ switch with 16 ports, each featuring 16 λs. As shown in the bottom part of Figure 3, the switch operates in 2 stages, each having the duration of 1 TF. In stage 1, data units belonging to a TF are received on each wavelength, separated by a WDM de-multiplexer (DMUX), and are aligned to the CTR by an alignment system, which also provides input buffering of data units until they can be switched. In stage 2 all the data units are transferred through the switching fabric (switched) to their corresponding output port. At the output port data units are transmitted on their selected λ through a WDM MUX. When immediate forwarding is realized, stage 2 takes place in TF(t+1) following TF(t) in which the data units were received and aligned. In non-immediate forwarding, stage 2 takes place in TF($t+N$), where $N>1$.

The switch architecture shown in Figure 3 does not require buffering of data units at the output ports. Moreover, as shown in Figure 4, alignment systems can be realized in a simple way with three first-in-first-out (FIFO) queues with mutually exclusive read and write access. The queue from which data units are retrieved for switching is changed in a circular manner at the beginning of each TF of the CTR. The queue, in which data units are stored as they are received from the input wavelength, is changed in a circular manner whenever the TF to which data units belong (i.e., the TF of the link's UTR) changes. As a result, the control of the alignment system is very simple and no memory access speedup is needed since a buffer is never read and written at the same time.

The switching control is simple because the fabric configuration is changed once per TF (e.g., 80,000 per second with 12.5 µs TFs), according to a *predefined and repetitive pattern*. The predefined switching pattern,

established at the FλP setup time, is such that output contention is avoided. As a result, no switching fabric speedup is needed (i.e., the rate of input/output connections through the switching fabric is the same as the transmission rate on input and output wavelengths). Moreover, as detailed in Section E, fractional λ switches can be based on blocking switching fabrics, such as Banyan networks [15].

D. Fractional λ Pipes and Multi-protocol Support

The repeated switching pattern of a fractional λ switch is determined when fractional λ pipes (FλPs) are created through a FλS network by reserving one or more TFs for each FλP in every time cycle or super cycle. Each FλP provides a constant bit rate pipe through which data units are transferred without any loss due to congestion and with delay jitter not greater than one TF.

FλS's pipeline forwarding does not rely on a specific data format since the TF can be regarded as a virtual container, the content of which is switched according to a predefined pattern. Consequently, FλPs provide switching and transport for various kinds of data units from both the packet switching world (e.g., IP packets, ATM cells, Ethernet frames) and the circuit switching world (e.g., SONET STS-1 frames). Data units of a different kind can coexist on the same communications channel during different TFs (i.e., carried over different FλPs).

Figure 5 shows a scenario in which multiple wavelength channels (λs) are multiplexed on the same optical link between two nodes. Multiple FλPs are defined over the three λs, some of them having reserved TFs on more than one wavelength channel[2]. Data units of different protocols, e.g., IP/MPLS, ATM, Frame Relay (FR), Fiber Channel (FC), and SONET, are multiplexed over each λ. Different protocols are carried within different FλPs, in other words, during different TFs.

E. FλP Blocking Problem

Given a link on the route of an FλP through the network, the identity[3] of the TFs reserved on the link is constrained by the identity of the TFs reserved on the upstream link. In other words, once the identity of a TF to be reserved on a link is fixed, the identity of the corresponding TF on each of the other links of the FλP path is uniquely determined. The selection of such TFs is based on the propagation delay on the links and the type of pipeline forwarding (e.g., immediate forwarding, non-immediate forwarding) performed by switches; this can be easily inferred by the forwarding example in Figure 2.

Reserving resources for an FλP requires solving a *scheduling* problem to find a feasible sequence of TFs, called a *schedule*, on the links on the route from source to destination. When a new FλP is being created and resources are being looked for, unavailable TFs on a link, i.e., already allocated to other FλPs, represent a constraint on reservations to be performed on the other links included in a path. Consequently, a reservation can fail even though enough capacity is available on all the links on that FλP's path. This will happen if the identity of the TFs on the various links does not match the requirements imposed by the deployed pipeline forwarding (e.g., immediate forwarding, non-immediate forwarding). In this case the FλP is said to be *unschedulable* or blocked in order to ensure deterministic service guarantees.

In asynchronous packet networks *admission control* performs heuristic checks on resource availability only: no explicit constraint in the time dimension does exist and unschedulability does not apply directly to asynchronous packet networks. However, resource utilization is well below 100 % (and often well below 50 %) and connections or calls are blocked when probabilistic service guarantees are to be provided.

The switching fabric deployed in fractional λ switches is another potential cause for an FλP setup to be impossible even though enough transmission capacity is available. A *blocking switching fabric* might be unable to provide a switching path from an idle inlet to an idle outlet due to multiple input/output paths contending for internal switching resources. In this case, an FλP cannot be accepted, i.e., it is *blocked*, even thought there are transmission resources on the corresponding optical channels.

Nonetheless, blocking switching fabrics are appealing due to their minimum complexity, and hence high scalability and low cost. For example, an N-by-N crossbar switch, which is strict-sense non blocking, contains N^2 crosspoints. However, an N-by-N Banyan switching fabric implemented, for example, by interconnecting $\log_2 N$ stages of $N/2$ 2-by-2 crossbar switches (4 crosspoints each) contains $2 \cdot N \cdot \log_2 N$ crosspoints. For example, a 10-fold increase in the required input/output capacity results in a 1,000-fold increase in the complexity and cost for a crossbar, but only a 30-fold increase for a Banyan switching fabric. Deployment of blocking switching fabrics is not practical in either circuit switches or packet switches.

In fractional λ switches the switching fabric configuration is changed each TF according to a predefined and reoccurring switching pattern every

---

[2] An FλP can use different wavelengths on the traversed optical links.
[3] The *identity* of a TF is its position in the time cycle.

cycle or super-cycle. Since the scope of contention on internal switching resources is limited to a single TF, blocking is eliminated during switching pattern computation by avoiding conflicting input/output connections during the same TF.

Deployment of blocking switching fabrics (e.g., Banyan) in fractional λ switches restricts the solution space of the FλP scheduling problem to the TFs during which at least one connection is possible from one of the λs of the input link to one of the λs of the output link on the path of the FλP. The suitability of Banyan switching fabrics for the implementation of fractional λ switches is confirmed by the simulation results reported in Section IV.

FλP blocking probability is defined as the probability for a resource reservation on an FλS network to fail even though enough resources (i.e., TFs) are available on all the links of the FλP's path. This can happen because of unschedulability and/or internal switch blocking. The FλP blocking probability depends on several parameters, such as the size of the time cycle, the number of optical channels per optical link, the degree of connectivity of the network (affecting the amount of alternative routes for a FλP), and the kind of switching fabrics deployed in fractional λ switches. The presented simulation study aims at evaluating the extent to which some of the above parameters affect the FλP blocking probability.

III. THE CALL-LEVEL SIMULATOR

In order to study the blocking probability in an FλS network, an event-driven call level simulator had been implemented. The simulator assumes that only one protocol is carried on the FλS network (i.e., data units from different sources can always be transmitted during the same TF) and does not currently support non-immediate forwarding.

Various types of sources generate resource requests (calls) according to a specific probabilistic arrival model. Calls that arrive at the FλS network's signaling controller are processed in order to determine whether to accept or reject them. This decision is based on two steps:

1. If resources are available within an existing FλP between the ingress and egress nodes of the FλS network through which the call is to be routed, the call is accepted and the occupancy of the TFs reserved to the FλP is updated according to the reservation request, i.e., number and size of data units required per time cycle.

2. Otherwise, a new FλP with suitable capacity is setup.

The objective of the simulations is to devise the *call blocking probability* as the ratio between the number of blocked calls and the total number of received calls. The *call* blocking probability is considered instead of the *FλP* blocking probability with the objective of giving a view of the network performance as close as possible to the one perceived by the end user. Whenever the simulation parameters are such that the capacity requirements of each call are as large as the FλP capacity granularity, the call blocking probability coincides with the FλP blocking probability since each call is carried by a dedicated FλP.

A. Architecture

The simulator, written in C++ to take advantage of the modularity offered by object-oriented programming, is based on the components described below.

- The *event scheduler* is the heart of the simulator; it picks the next event from an *event queue*, which is sorted by increasing event due time. An event can be one of two kinds: the *arrival or setup* of a call and the *clearing or teardown* of a call.

  When processing a call arrival, the event scheduler checks whether the call can be accepted—which involves TF scheduling. In the case of a positive response it reserves the proper network resources (i.e., it updates the occupancy information of the FλP carrying the call) possibly after having created the FλP.

  A call clearing event causes all the resources reserved for the call to be released. In the case that the FλP carrying the call is not being used by any other call, the FλP is torn down and its reserved TFs are released.

- *Call sources* generate calls characterized by bandwidth, data unit size, destination and duration. The simulator provides a large variety of call sources; however, due to the very high capacity of the communications links, the presented study deploys only Video on Demand (VoD) call sources with a 2 Mb/s bandwidth requirement. The VoD call arrival is modeled as a Poisson process and the duration is obtained from a gamma distribution function with a 2 hour maximum.

  The current version of the simulator determines the path of a call, and hence of the FλP intended to carry it, according to a route previously configured between the source and the destination. Obviously, deployment of multiple alternative routes (not supported by the current version of the simulator) would result in lower blocking probability.

- In order to ensure statistically meaningful results, a

statistical module is used to determine the end of both the initial transient phase and the simulation.

B. Scheduling

Scheduling is performed by the event scheduler whenever a FλP is to be created as a consequence of call event processing. Given the intended capacity for the FλP, the scheduler devises the equivalent number of TFs per time cycle to be reserved. Then, the scheduler executes a *distributed scheduling algorithm* derived from that presented in [1] for the ATM switching.

Next this section will describe the implemented scheduling algorithm in two steps. First, an algorithm for networks that do not use WDM, i.e., each communications link features a single communication channel, is presented. The algorithm is a variation of the one presented in [1], modified to take into account a blocking switching fabric within fractional λ switches. Support for multiple channels per communication link is then added to the scheduling algorithm.

B.1 Single Channel per Communication Link

Scheduling and resource reservation are based on a data structure called an *availability vector*, which has size of k bits–one bit for each TF in the time cycle. As shown in Figure 6, scheduling employs the following availability vectors:

- A *link availability vector* is associated with each link of the network and contains the bit map of the TFs that have not yet been reserved.

- A *switch availability vector* is associated with each input/output pair of the network nodes. It contains the bit map of the TFs during which a connection can be established between the input/output pair, given the existing input/output connections through the switching fabric during each TF.

- When the scheduler processes an FλP setup, it generates an *FλP availability vector* that will eventually contain the bit map of the TFs that can be reserved for the FλP. Resource allocation is performed by selecting the needed number of TFs among those tagged as available in the FλP availability vector bit map.

Figure 6 shows an example of the computation of an FλP availability vector; the labels on a link represent the delay, in TFs, between (the egress of the alignment systems in) the nodes at its ends[4]. The FλP availability vector is initialized to the link availability vector of the first link on the path of the FλP, as shown by the initial FλP availability vector in Figure 6. Then, the FλP availability vector is cyclically shifted to the right a

number of times equivalent to the link label. A bit-by-bit logical AND operation is performed between the shifted (interim) availability vector, the availability vector of the next link on the path, and the switch availability vector of the input/output pair to which the two links are connected. The resulting bit vector is shifted until the (final) FλP availability vector reaches the FλP egress point.

If the (final) FλP availability vector contains enough TFs, the FλP is accepted, the required number of TFs is chosen by the scheduler, and resources are reserved on all the links and switches on the path by updating the link and switch availability vectors according to the chosen TFs.

The set of TFs chosen by the scheduler is called a *schedule*. The choice of the schedule when there is more than one possibility affects service provided by the network (in terms of FλP access delay and its variation) and achievable network utilization, and ultimately, the blocking probability. However, analysis of the implications of schedule choice is outside the scope of this work and left for further study.

B.2 Multiple Channels per Communication Link

The availability of multiple channels on each communication link between fractional λ switches adds a degree of freedom to the scheduling problem. If all the switching fabrics are non-blocking and full wavelength conversion is possible at every network node (i.e., a TF received on any incoming wavelength can be transmitted on any outgoing wavelength), the scheduling algorithm presented in the previous section can be applied. A bit of the availability vector of an output link is set if the corresponding TF is available on at least one of the link's wavelengths.

If at least one switch has limited wavelength conversion capability or a blocking switching fabric, the switch availability vector captures both the constraints. A variation of the scheduling algorithm must be deployed in order to individually take into account multiple alternative input/output connections through the switch for routing an FλP. Additional interim FλP availability vectors are generated at each node, as shown in Figure 7, with the solution space growing accordingly.

Figure 7 shows an example of computation of a set of FλP availability vectors on a network with two optical channels per link. Two *channel availability vectors* are associated with each link and a *three-dimensional switch availability vector* is associated with each node. The switch availability vector has 8-by-2-by-2 elements (bits), with each one indicating the feasibility of a specific input channel/output channel connection during

---

[4] This delay accounts for the propagation delay through the switching fabric and the optical link, and the receiving end alignment time.

a TF.

The initial set of (two) FλP availability vectors is initialized to the availability vectors of the two channels on the first link. At the next node, a three-operand bit-by-bit logical AND operation is performed among all the possible combinations of the following:

1. One of the shifted (interim) availability vectors;
2. One channel availability vector of the next link on the path;
3. The one-dimensional switch availability vector of the switching fabric connection between the inlet and outlet connected to the selected input channel and output channel, respectively.

The above operation yields four availability vectors, namely, as many as the product of the number of channels on the two links. The resulting link availability vectors are shifted and combined with the switching and channel availability vectors at the following nodes until a (final) set of 16 FλP availability vectors is produced at the FλP egress point.

If at least one of the (final) FλP availability vectors contains enough TFs, the FλP is accepted, the required number of TFs is chosen by the scheduler based on the bit map in one of the final FλP availability vectors, and resource reservation is performed on all the links and nodes on the path by updating the channel availability vectors and the switch availability vectors according to the chosen TFs.

B.3 Scheduling Complexity and Heuristics

The complexity of the FλP availability vector computation process is on the order of

$$\sum_{n=0}^{N}\prod_{l=0}^{n} c_l \ ,$$

where c_l is the number of channels on link l and N is the number of nodes on the path of the FλP. If the number of channels per link is constant ($c_l = C, \ \forall l, \ 0 \le l \le N$), the complexity of the scheduling algorithm is exponential in number of channels per link and nodes per path. Therefore the presented scheduling algorithm has poor scalability and some heuristics is used to limit the complexity as the network size grows.

In the example depicted in Figure 7, the third FλP availability vector between nodes B and C indicates that there is no TF availability on the upstream portion of the path. Thus, processing this FλP and derived FλP availability vectors on the downstream portion of the path (the third and eleventh FλP availability vectors of the last link, in the example in Figure 7) is useless. A first improvement of the algorithm's scalability stems from avoiding processing FλP availability vectors whose elements are all zero. This approach is generalized by

processing, at each step, only "promising" interim FλP availability vectors. Our simulator limits the maximum number of FλP availability vectors considered at each step of the algorithm. After having performed the bit-by-bit logical AND operation, only the *M* most "promising" interim FλP availability vectors are considered for the next computation step. Different criteria can be used for ranking FλP availability vectors according to how "promising" they are. The chosen criteria is likely to affect the effectiveness of the scheduling algorithm and consequently, the FλP blocking probability. However, the evaluation of the ranking criteria is outside the scope of this work and left for further study. Our simulator ranks FλP availability vectors according to the number of available TFs they contain. Such a scheduling algorithm does not ensure finding a schedule whenever it exists, but it provides for easy implementation that is good enough.

IV. SIMULATION RESULTS

The simulations shows the impact of blocking on the efficient use of bandwidth or utilization; the concern being that blocking will prevent communications resources from being fully used. The objective of this performance study is to evaluate how various network parameters, such as number of time frames per time cycle and number of channels per optical link, affect the blocking probability, and to provide guidelines for minimizing the blocking.

The results of the simulations are presented in graphs that plot the blocking probability of calls routed through a link versus its utilization. The blocking probability is calculated by the simulator as the number of rejected resource reservation requests over the total number of requests.

The efficiency of FλS is provided by the highest utilization that is achievable with negligible blocking probability. In fact, whenever blocking takes place, the blocking probability grows quickly as the call load increases and link utilization approaches 100 %. Since there is no interest in operating a network at a load level in which resource reservations are rejected at high rates, the most significant part of the plots is where the blocking probability becomes non-null and starts growing.

In order to isolate separate factors affecting blocking probability, simulation results for a single switch are first shown in Section IV.A, and results for a network of switches are then presented in Section IV.B.

Various system parameters are changed throughout the study in order to assess their impact on blocking probability, while the following settings are common to all simulations.

- The time cycle duration is 12.5 ms; given the number of time frames per time cycle, the time frame duration is adjusted accordingly. For example, when the time cycle contains 1000 time frames, each time frame is 12.5 µs.

- The link capacity is 40 Gb/s. Since simulations run with different numbers of channels per link, and channel capacity changes accordingly. For example, when 4 channels are used on a link each one has a capacity of 10 Gb/s.

- Call traffic consists of video-on-demand sessions that require a bandwidth of 2 Mb/s. The main factor in determining blocking probability is the number of time frames per time cycle required to satisfy each call request. Such number ultimately depends on the ratio between call bandwidth requirement and link capacity. A call bandwidth of 2 Mb/s was chosen as realistic with respect to the capacity of the links used in our simulation experiment[5]. Deployment of various types of sources in order to experiment with multiple call arrival and call duration distributions was not considered to provide significant additional insight in the issues tackled by this work. In fact, the objective of this work is assessing the maximum network utilization achievable with negligible blocking, not evaluating the scheduling algorithm (e.g., its complexity or response time) or network dimensioning.

- Time driven switches have four input links and four output links. Since the number of channels per link is changed, the switching fabric size changes accordingly. However, since the overall link capacity is kept constant (40 Gb/s), the switching fabric capacity remains constant (160 Gb/s).

A. Single Node

Since the source and destination of each call are connected to the same switch, call blocking is due exclusively to the blocking nature of the switching fabric (no scheduling problem needs to be solved on a single switch).

Simulation results are presented in this section by plotting the blocking probability measured on one of the output links of the switch versus the utilization measured on the same link.

A.1 Single Channel per Link

When WDM is not used, i.e., each link has a single channel; a single switching fabric inlet/outlet is connected to a communications link. In this scenario, a call source is connected to each of the 4 inlets and a call sink to each of the 4 outlets.

In the simulations presented in this section the traffic loading the switch is balanced, i.e., uniformly distributed among all the inputs and the outputs. In other words, the destination of the calls generated by each source is chosen according to a uniform distribution among the call sinks connected to the 4 output ports, as shown in Figure 8. Consequently, the source of the traffic received by each destination is uniformly distributed among all the end systems connected to the 4 inputs.

Figure 9 shows the results of a set of simulations aimed at assessing how the number of time frames per time cycle impacts the blocking probability. Since the time cycle duration is 12.5 ms in all of the simulation runs and the number of time frames varies from 1 to 1000, the time frame duration is changed accordingly from 12.5 ms to 12.5 µs, respectively. The results clearly show that the time dimension of FλS compensates for the limitations in the space dimension of a Banyan interconnection network by reducing, and quite possibly virtually eliminating the consequent blocking.

The curve corresponding to the deployment of 1 time frame per time cycle represents a scenario in which each link contains only one FλP whose capacity is the channel capacity, i.e., an optical cross connect scenario. An input/output connection is statically allocated to the traffic between the respective ports, even if the overall traffic is not enough to fill it up. Calls from the same input to other outputs are rejected even though both the input and output links have unused capacity. As expected, due to the uniform distribution of call destinations and as confirmed by the simulation results, the probability for incoming calls to be blocked is 75 %.

Deployment of at least 4 time frames per time cycle enables 4 FλPs to be set up through each of the inlets and outlets, i.e., the exact number of FλPs needed to provide connectivity between each source-destination pair. However, due to the blocking nature of the switching fabric and the unpredictable call arrival pattern, FλPs between all source-destination pairs are not necessarily feasible. Consequently, the blocking probability is always higher than 25 %, even when the call load is fairly low.

As the number of TFs per time cycle increases, the flexibility in the selection of TFs for the creation of FλPs increases and the impact of the blocking switching fabric on the overall efficiency is reduced. With a large number of time frames per time cycle, e.g., 1000, the blocking probability is virtually zero for utilization up to 97 %.

---

[5] We expect the network operator to dimension links according to the bandwidth of the calls routed on them. In fact, in order for the network to be operated under a good level of traffic multiplexing the number of calls, hence users, sharing each link should not be too small.

A.2 Multiple Channels per Link

The wavelength dimension, as well as the time dimension, is expected to impact on the blocking characteristics of a fractional λ switch. In fact, a Banyan network provides only one path between each inlet/outlet pair, but when each input link has two channels—each one connected to an inlet—there are four possible paths between each input link/output link pair. Even if some of these four paths might share switching resources (elementary switching elements or connections among them), the probability that an input/output path be feasible for an arriving call is intuitively larger than without WDM. For example, in the one channel per link configuration depicted in Figure 10 (a) input 1 cannot be connected to output 1 while input 5 is connected to output 2. Instead, in the example shown in Figure 10 (b) of two channels per link these two connections are feasible at the same time.

As it can be seen in Figure 11 and Figure 12, increasing the number of channels per link from 1 to 16 results in a lower blocking probability for both time cycle configurations (1 TF per time cycle and 4 TFs per time cycle).

The results shown in Figure 11 refer to a scenario in which each channel contains only one FλP whose capacity is the channel capacity, i.e., a lambda switching (also called lambda routing) scenario. The effect of the blocking nature of the Banyan switching fabric is mitigated by the availability of enough channels per link. The blocking due to the switching fabric is eliminated (or made negligible) by increasing the number of time frames as shown by Figure 12 and Figure 13.

B. Network of switches

Setting up an FλP across multiple switches requires solving a scheduling problem to identify the proper time frames to be reserved on each traversed link. As discussed in Section II.E, this can result in blocking. Thus, in a network scenario, blocking encompasses the effect of both scheduling and blocking switching fabric deployment.

In order to provide a quantitative assessment of the efficiency of FλS, a network utilization index is required. For each network scenario used in the simulations, the link traversed by the largest amount of call traffic is identified as the *bottleneck link*. The bottleneck link's utilization is used as an index of the utilization of the overall network. The results of the simulations are presented in graphs that plot the aggregate blocking probability of calls routed through a link versus the bottleneck's link utilization.

Two network topologies are used for the simulations.

A symmetric one (Section B.1) allows a better understanding of the traffic distribution and hence a simpler interpretation of the simulation results. An asymmetric topology (Section B.2) provides a scenario closer to real world deployment. In both the symmetric and asymmetric scenarios, the time cycle contains 1000 time frames.

B.1 Symmetric Topology

Figure 15 shows the blocking probability measured on one of the output links of switch G[6] operating in the network depicted in Figure 14. Full dots represent call sources, while empty dots represent call sinks. The destination of calls generated by each source are uniformly distributed among all the call sinks. Thus, the network and call traffic loading it are symmetric.

Comparing the curves presented in Figure 15 with those devised in a single switch scenario with 1000 time frames per time cycle and presented in Figure 9 and Figure 13 shows the blocking increase due to scheduling. However its contribution for all practical purposes is negligible. For example, with 4 channels per link, the maximum utilization without blocking achieved on a single switch is above 97 %; while on switch G's output links maximum utilization is slightly above 91 %. In other words, the combination of multiple WDM channels per link and a large number of time frames per time cycle enables a network of Banyan-based fractional λ switches to achieve very high utilization with negligible blocking.

Fractional λ switches based on a non-blocking crossbar switching fabric were used in a new set of simulations, in order to isolate the contribution of TF scheduling from that due to Banyan network switching fabric. As shown by the results presented in Figure 16, the utilization achievable without blocking is significantly higher only in the single channel configuration. In other words, the impact of Banyan switching fabrics on call blocking can be essentially neglected when multiple channels per link are deployed.

B.2 Asymmetric Topology

The network topology depicted in Figure 17 is obtained by trying to break the symmetry of the topology (Figure 14) studied in Section B.1. Multiple paths exist between a few of the call source (full dot)-call sink (empty dot) pairs. Corresponding calls are also routed over the longer path(s) in order to enable the evaluation of blocking of long distance calls, on which scheduling is more complex.

The graphs in Figure 19 plot the blocking probability measured on the links highlighted in Figure 17 versus

---

[6] Due to the symmetry of the network and call traffic, the blocking probability is basically the same on all switch G's output links.

the utilization of the most loaded link within the network (bottleneck link). Given the topology and the distribution of the source-sink pairs, the bottleneck link changes depending on the actual call arrival process. Thus, at the end of every simulation the most loaded link will have been devised and used as an indication of network utilization. With 2 or more channels per link, utilization of at least 80 % with negligible blocking probability is achieved on all the considered links.

V. DISCUSSION

This paper presents *fractional λ switching* (FλS) and a simulation study of call blocking in FλS networks. FλS uses a global *common time reference* (CTR), which is realized with UTC, to implement *pipeline forwarding* (PF) of time frames; each time frame is a virtual container with 5-20 Kbytes. Since the time frames boundaries are explicitly identified, the CTR accuracy should be less than one half of a time frame. Pipeline forwarding, over a meshed FλS network, requires that the delay between any two switching fabric inputs be an integer number of time frames. This is ensured by an *alignment to CTR* operation before each switching fabric input.

Simulation results show that blocking does not compromise the efficiency of FλS: in most cases studied, network utilization above 90% is achieved with negligible blocking probability—even when switching fabrics are based on Banyan networks. Thus, given the simplicity and scalability of fractional λ switches, FλS represents the most inexpensive solution for *efficient* and *deterministic bandwidth provisioning* in WDM networks.

The efficient and deterministic bandwidth provisioning with no loss and small constant jitter of FλS enables a network "**nirvana**" with IP/MPLS, as shown in Figure 18. The fractional λ pipes (FλPs) that are realized in FλS network have the same deterministic characteristics as leased lines in SONET and circuit emulation in ATM with several advantages: (*i*) IP/MPLS packets are transferred through a FλS network with no format change, (*ii*) simple aggregation or grooming in the time domain is possible, (*iii*) IP/MPLS header processing is performed only at the edges, and (*iv*) FλS is uniquely suitable for all-optical switching [2].

ACKNOWLEDGMENTS

REFERENCES

[1] M. Baldi and Y. Ofek, "End-to-end Delay of Videoconferencing over Packet Switched Networks," IEEE/ACM Transactions on Networking, Vol. 8, No. 4, Aug. 2000, pp. 479-492.

[2] M. Baldi, Y. Ofek, "Realizing Dynamic Optical Networking," Optical Networks Magazine, Special Issue "Dynamic Optical Networking: around the Corner or Light Years Away?", September 2003.

[3] C-S. Li, Y. Ofek, A. Segall and K. Sohraby, Pseudo-Isochronous Cell Switching in ATM Networks, *IEEE INFOCOM'94*, pp. 428-437, 1994.

[4] C-S. Li, Y. Ofek, A. Segall and K. Sohraby, "Pseudo-Isochronous Cell Forwarding," Computer Networks and ISDN Systems, 30:2359-2372, 1998.

[5] M. Baldi, Y. Ofek and B. Yener, "Adaptive Group Multicast with Time-Driven Priority," IEEE/ACM Transactions on Networking, Vol. 8, No.1, Feb. 2000, pp. 31-43.

[6] Y. Ofek and M. Faiman, "Distributed Global Event Synchronization in a Fiber Optic Hypergraph Network," The 7th International Conference on Distributed Computing Systems, Berlin, September 1987.

[7] Y. Ofek, "The Topology, Algorithms and Analysis of a Synchronous Optical Hypergraph Architecture," Ph.D. Dissertation, Electrical Engineering Department, University of Illinois at Urbana, Report No. UIUCDCS-R-87-1343, May 1987.

[8] D. K. Hunter and D. G. Smith, "New architectures for optical TDM switching," IEEE/OSA Journal of Lightwave Technology, vol. 11, no. 3, pp. 495-511, Mar. 1993.

[9] I. P. Kaminow et al., "A wideband all-optical WDM network," IEEE Journal on Selected Areas in Communications, vol. 14, no. 5, pp. 780-799, June 1996.

[10] P. Gambini et al., "Transparent optical packet switching: network architecture and demonstrators in the KEOPS project," IEEE Journal on Selected Areas in Communications, vol. 16, no. 7, pp. 1245-1257, Sept. 1998.

[11] Nen-Fu Huang, Guan-Hsiung Liaw, and Chuan-Pwu Wang, "A novel all-optical transport network with time-shared wavelength channels," IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, pp. 1863-1875, Oct. 2000.

[12] H. F. Jordan, D. Lee, K. Y. Lee, and S. V. Ramanan, "Serial array time slot interchangers and optical implementations," IEEE Transactions on Computers, vol. 43, no. 11, pp. 1309-1318, Nov. 1994.

[13] R. Srinivasan, Arun K. Somani, "A Generalized Framework for Analyzing Time–Space Switched Optical Networks," IEEE Journal on Selected Areas in Communications, vol. 20, no. 1, pp. 202-215, January 2002.

[14] Suresh Subramaniam, Eric J. Harder, and Hyeong-Ah Choi, "Scheduling Multirate Sessions in Time Division Multiplexed Wavelength-Routing Networks," IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, pp. 2105-xxx, October 2000.

[15] Bo Wen and Krishna M. Sivalingam Routing, "Wavelength and Time-Slot Assignment in Time Division Multiplexed Wavelength-Routed Optical WDM Networks," IEEE INFOCOM 2002.

[16] Y. Ofek, "Generating a Fault Tolerant Global Clock using High-speed Control Signals for the MetaNet Architecture," IEEE Transactions on Communications, May 1994.

[17] L. R. Goke, G. J. Lipovski, "Banyan Networks for Partitioning Multiprocessor Systems,"1st Annual Symposium on Computer Architecture, Dec. 1973, pp. 21-28.
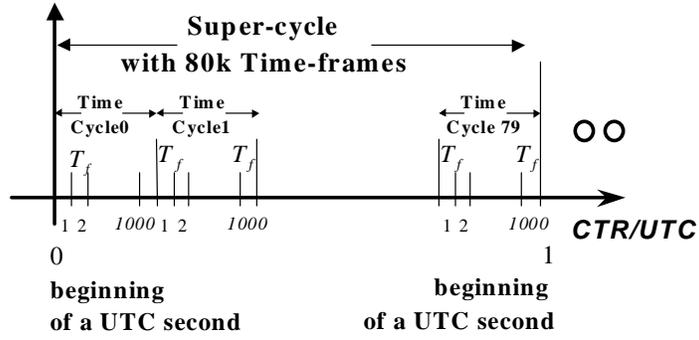
Figure 1: The common time reference (CTR) with T_f=12.5 μs
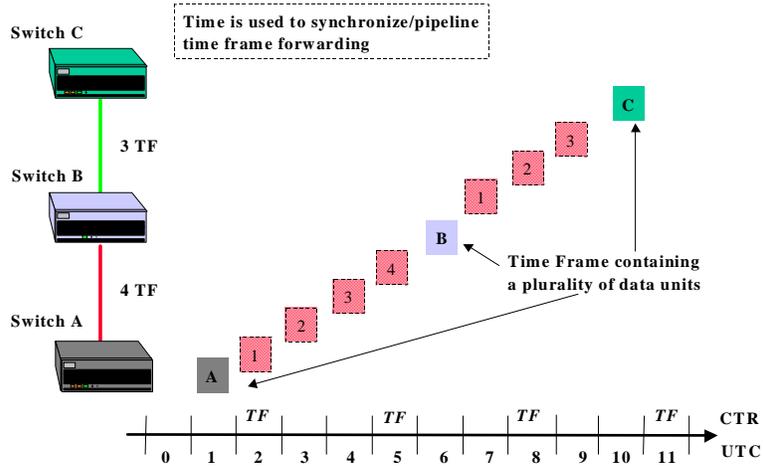

Figure 2: FλS with immediate forwarding

12
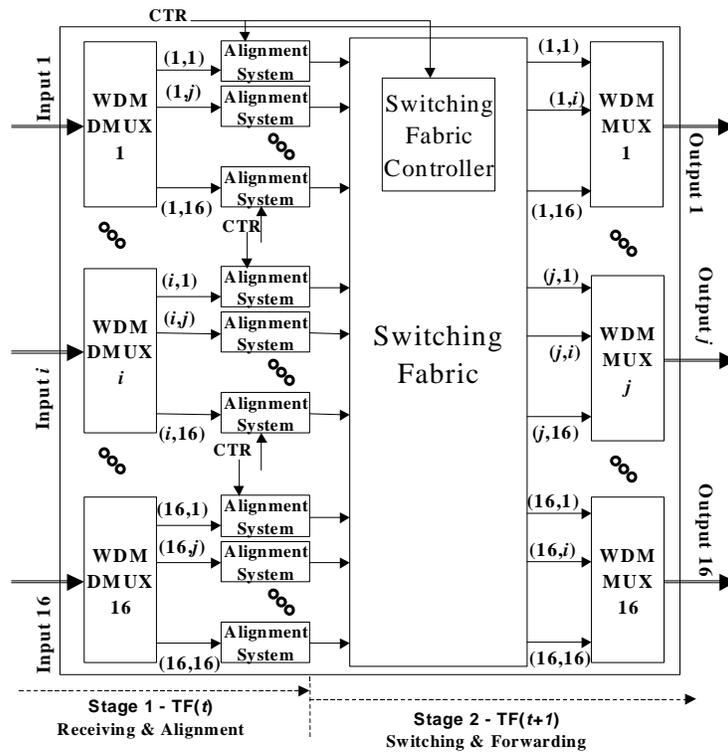
Figure 3: Fractional λ switch architecture and operating stages
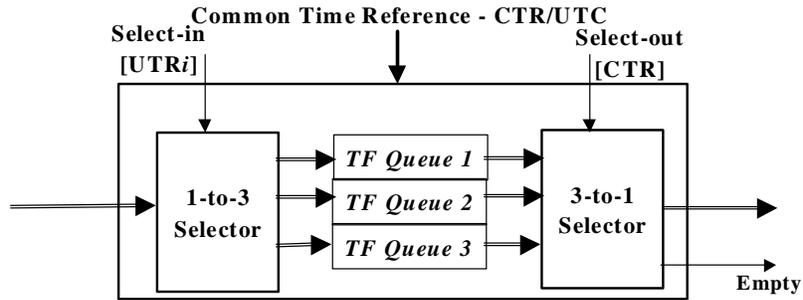

Figure 4: Alignment System
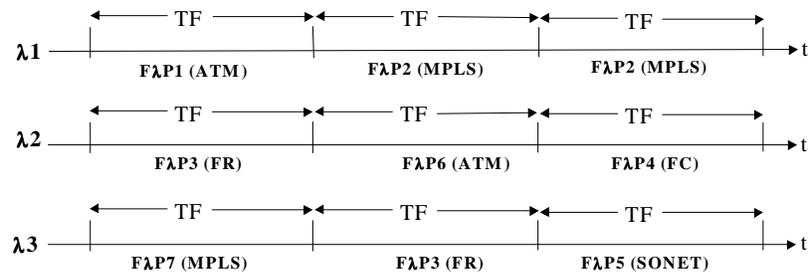
Figure 5: Multi-protocol support
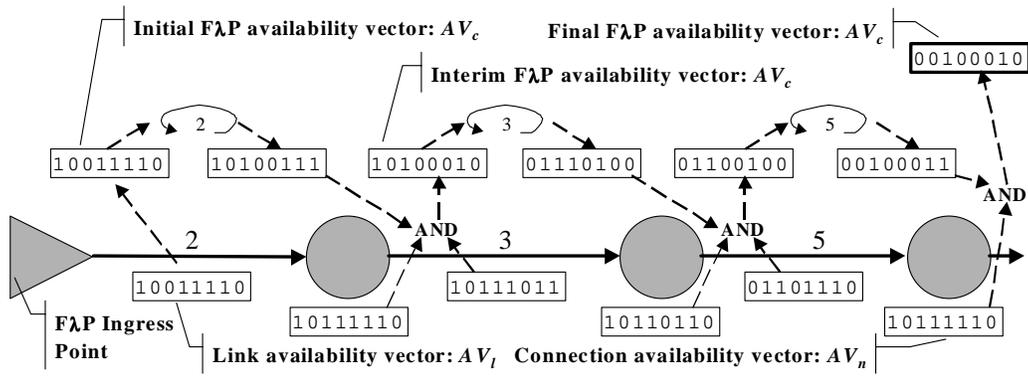
Figure 6: Computation of an FλP availability vector
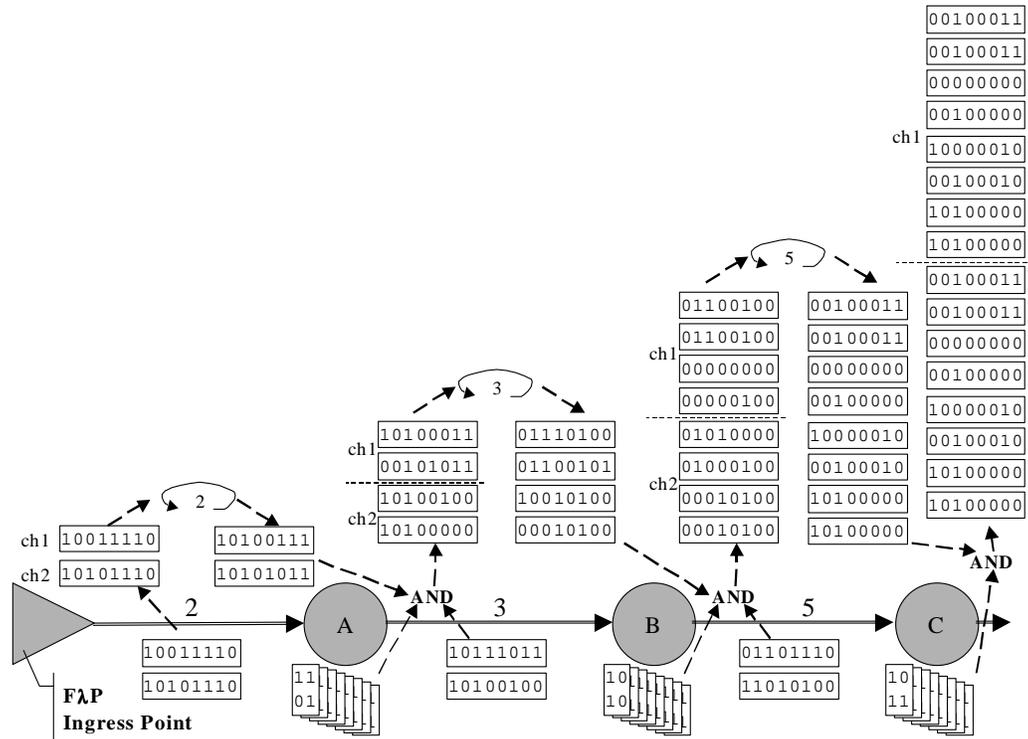
Figure 7: Computation of an FλP availability vector in a network using WDM with limited wavelength conversion capability or partial input/output channel connection capability
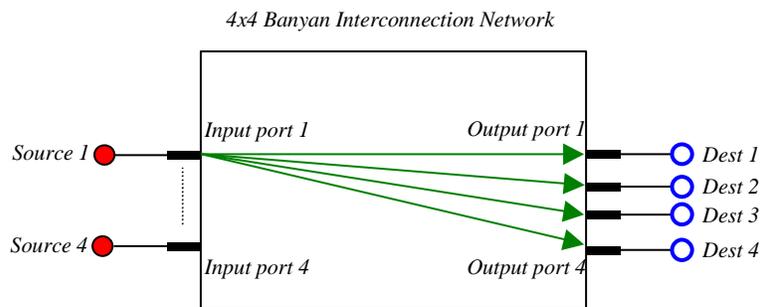

Figure 8: Balanced load call distribution
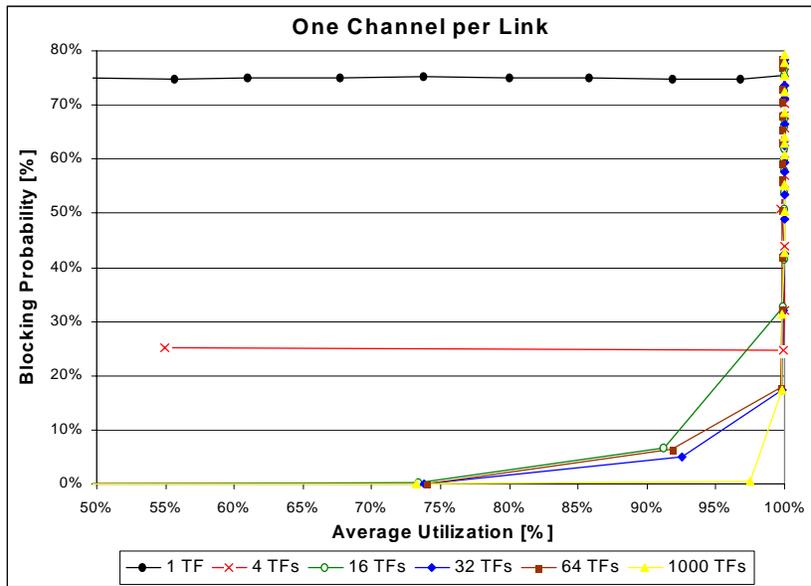
Figure 9: Impact on blocking probability of number of TFs per time cycle
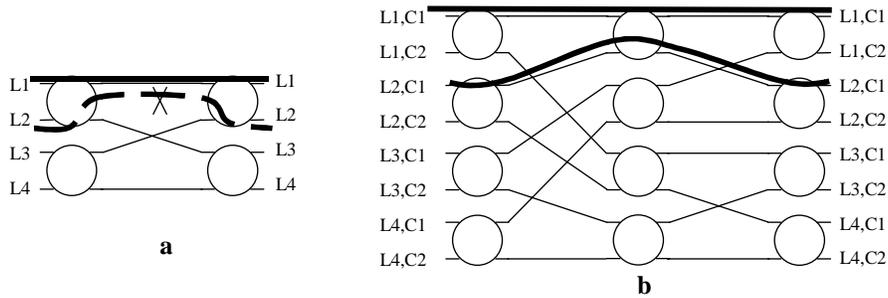
Figure 10: Impact of the number of channels per link on blocking
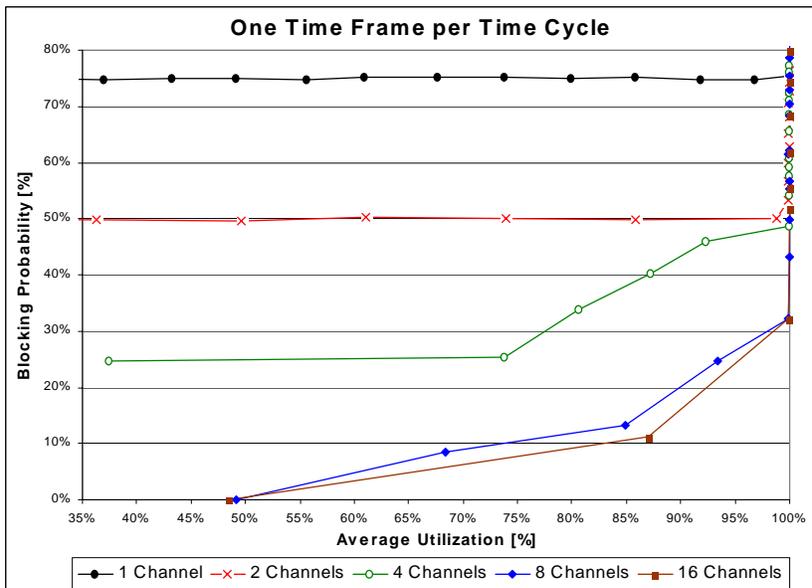
Figure 11: Impact on blocking probability of number of channels per link—1 TF per time cycle
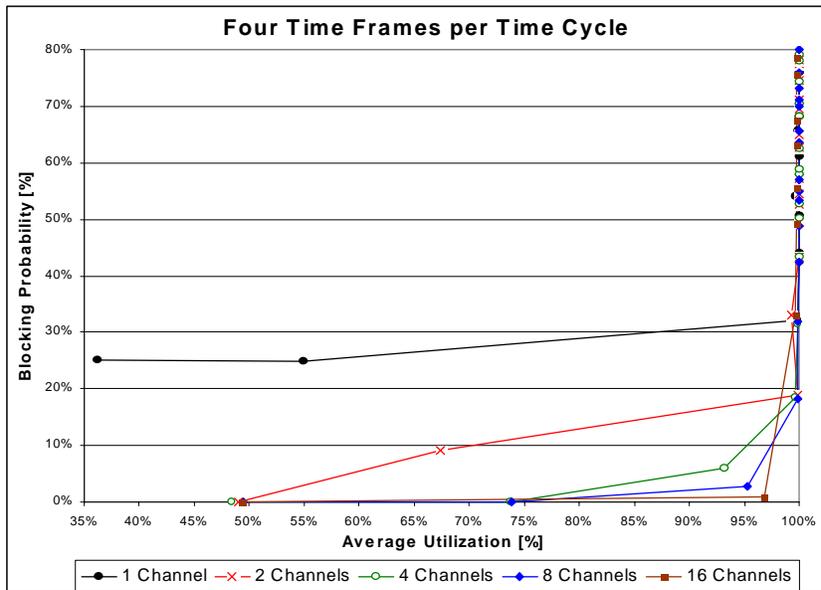
Figure 12: Impact on blocking probability of number of channels per link—4 TFs per time cycle
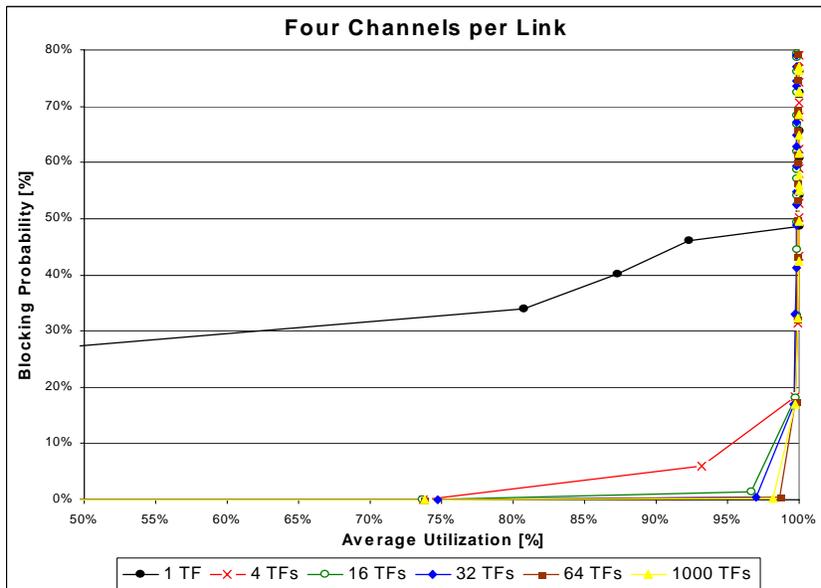
Figure 13: Impact on blocking probability of number of channels per link and number of TFs per time cycle
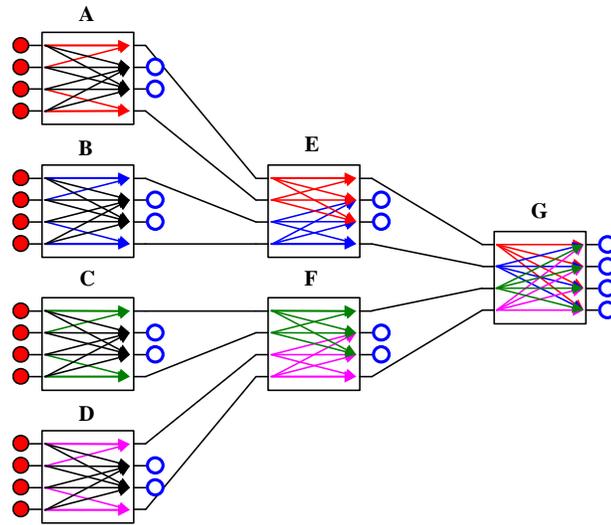
Figure 14: Symmetric network topology
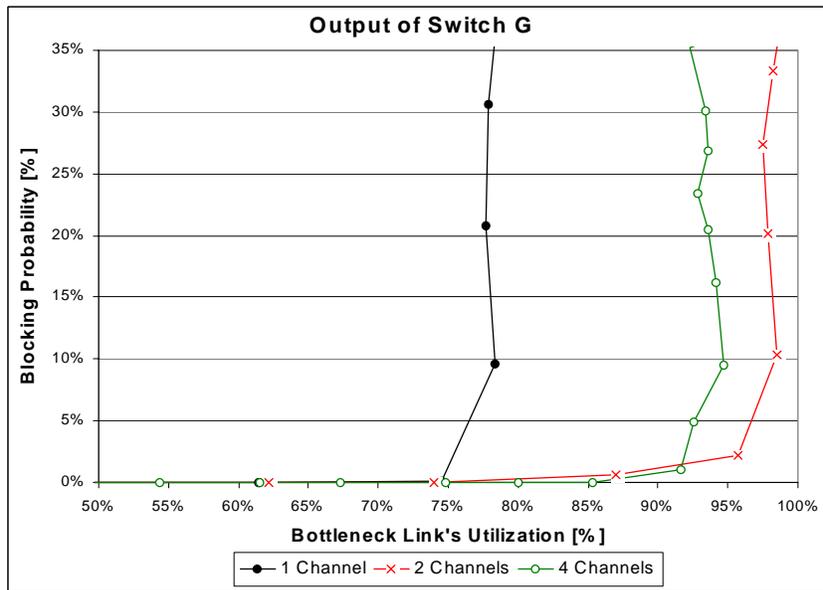
Figure 15: Output link of Banyan-based switch G

Output of Switch G
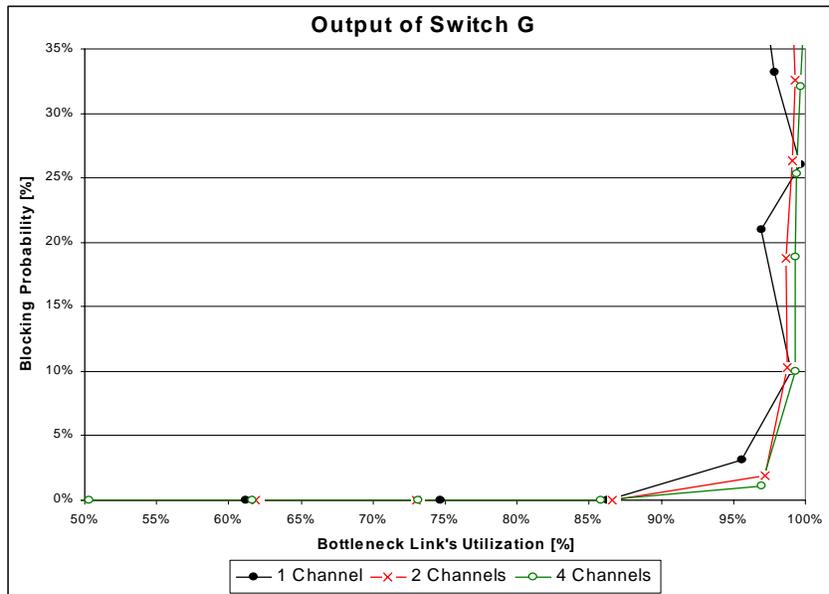
Figure 16: Output link of crossbar-based switch G
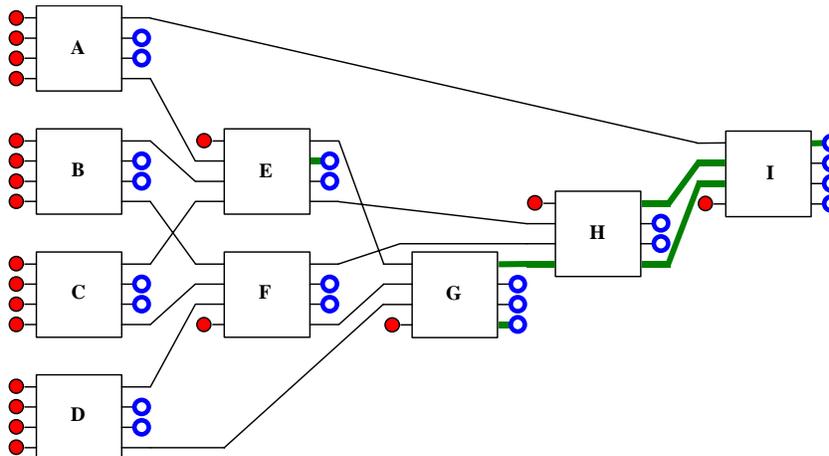
Figure 17: Asymmetric network topology

Packets all-the-way with header processing only at the edges
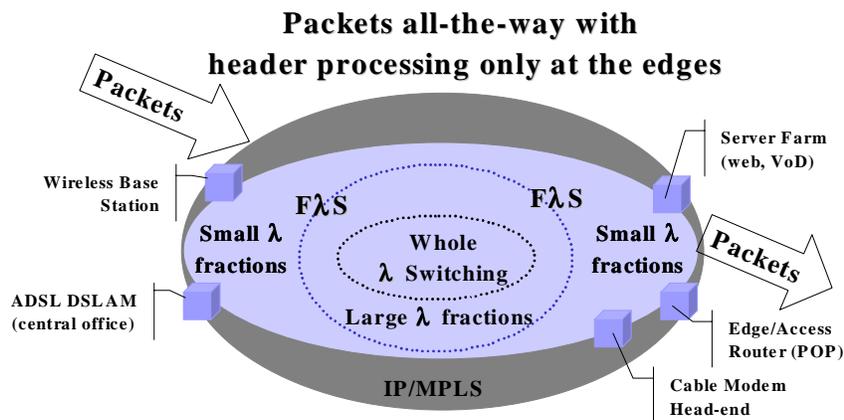
Figure 18: Convergence to an all-packet network with FλS while eliminating SONET
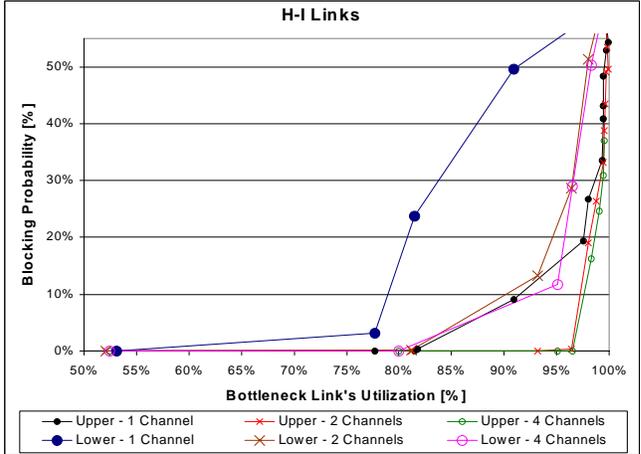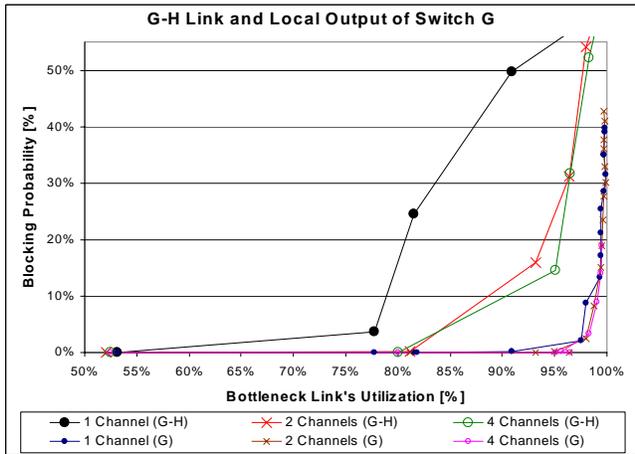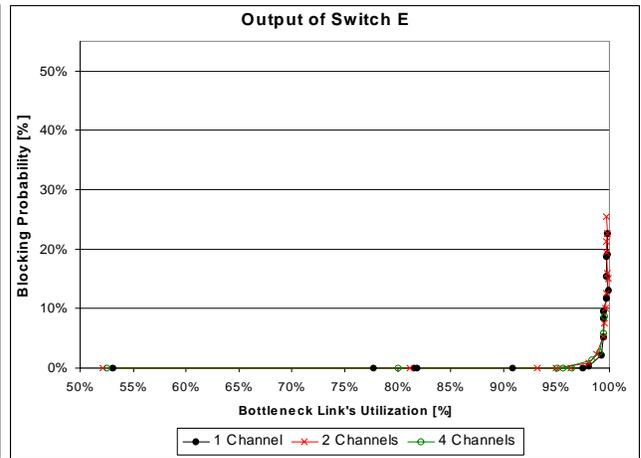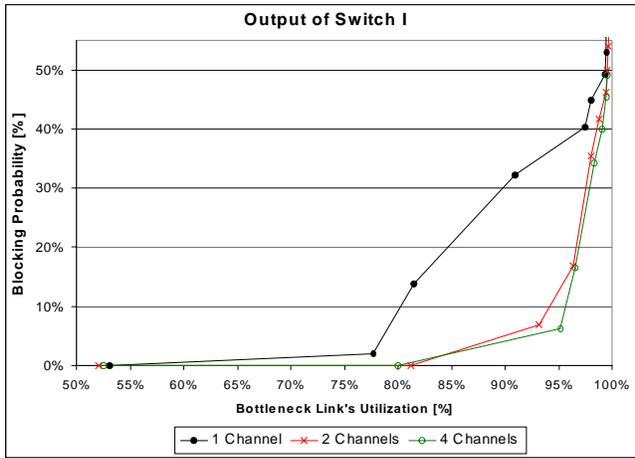
Figure 19: Simulation results on asymmetric topology network