Just-In-Time Signaling for WDM Optical Burst Switching Networks

John Y. Wei, *Member, IEEE,* and Ray I. McFarland, Jr., *Member, IEEE*

Invited Paper

***Abstract*—We describe the architecture, design, and implementation of a novel just-in-time (JIT) signaling protocol for optical burst switching (OBS) in wavelength division multiplexed (WDM) optical networks. The JIT-OBS paradigm is designed for ultra-low-latency unidirectional transport of data-bursts across an optical network. It combines the desirable features of circuit-switching and packet-switching. It features out-of-band control signal processing that eliminates buffering of data-burst at intermediate nodes, while minimizing the setup time, and maximizing the cross-connect bandwidth efficiency.We motivate and describe the architecture of JIT signaling, and analyze its basic performance. We present the detailed signaling message design and discuss the rationale and considerations that went into this design. We describe and examine various scenarios that illustrate the operations of the JIT signaling protocol (JIT-SP) in connection establishment and teardown. Finally, we describe and summarize the JIT signaling software prototype in the Washington, DC Testbed network implemented under the MONET project.**

***Index Terms*—Circuit switching, just-in-time signaling, optical burst switching, optical network, packet switching, signaling protocol, wavelength division multiplexing (WDM).**

I. INTRODUCTION

THE emergence of broad-band communications has increased the needs of bulk transport of high capacity signals and services. Multiwavelength reconfigurable optical networks offer such a capability beyond current transport technologies such as SONET and Gigabit Ethernet. The multiwavelength optical networking program (MONET) [1]–[4] sponsored by the U.S. Government's Defense Advanced Research Project Agency (DARPA) is a research consortium aimed at addressing the technology, architecture, and the management and control issues [5]–[8] for this new emerging technology.

In this paper, we describe a novel just-in-time (JIT) signaling protocol for optical burst switching (OBS) in wavelength division multiplexed (WDM) optical networks developed under MONET. The JIT-OBS paradigm is designed for ultra-low-latency unidirectional transport of data-bursts across an optical network. It combines the desirable features of circuit-switching and packet-switching, and features out-of-band control signal processing that eliminates buffering of data-burst at intermediate nodes, while minimizing the setup time, and maximizing the cross-connect bandwidth efficiency.

The MONET JIT Signaling Protocol (JIT-SP) is a signaling protocol designed specifically for WDM optical switching technology. JIT-SP is designed such that it can also be used for conventional circuit-switched connection signaling. It has been implemented and will be tested and validated in the MONET Washington DC testbed network.

This paper is organized as follows. In Section II, we provide some historical perspective on optical switch signaling developments and comment on other similar work reported in the research literature. In Section III, we outline a generic WDM switch architecture used for subsequent discussion. In this section, we describe the different WDM switching paradigms, motivating and introducing the JIT-OBS paradigm we developed. We also present an analysis of the basic performance of JIT Signaling. Section IV then describes the design assumptions and objectives used in the development of the MONET JIT Signaling Protocol. Section V presents several scenarios that detail the message dialogs between network nodes as JIT signaling is used to establish and tear down connections. Section VI describes the protocol message design detailing their syntax and semantics. In Section VII, we describe and summarize the MONET JIT-SP prototype implementation. Finally, Section VIII concludes the paper, and comments on the WDM switching experiments to be performed in the MONET Washington, DC testbed.

II. HISTORICAL PERSPECTIVE AND RELATED WORK

The advent of optical technology enables networks with bandwidth that are many orders of magnitude larger than existing ones. Such drastic change naturally requires us to rethink some of our networking fundamentals. Kleinrock observed in [9] that while a file transfer of size 1 Mb at 64 kb/s on a coast to coast connection in the United States consisted of 1000 U.S. spans (that is, the total amount could fit in the equivalent of 1000 pipes across the United States), and even at 1.644 Mb/s

---

Manuscript received June 9, 2000; revised October 4, 2000. This work was prepared through collaborative participation in the U.S. Government Multi-Wavelength Optical Networking (MONET) Project under Contract MDA 972-95-3-0027. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

J. Y. Wei is with the Telcordia Technologies, Inc., Navesink Research and Engineering Center, Red Bank, NJ 07701 USA (e-mail: wei@research.telcordia.com).

R. I. McFarland, Jr., is with the Department of Defense, Laboratory for Telecommunications Science, Ft. Meade, MD 20755-6911 USA (e-mail: rimcfar@lts.ncsc.mil).

Publisher Item Identifier S 0733-8724(00)10990-9.

the connection constituted 40 U.S. spans; at 1.2 Gigabit per second (Gb/s) the communication only constituted .05 U.S. Spans, or in other words, the entire file in transmission would be wholly contained with the boundaries of a single state that it was passing through. As the possible data rates enabled by optical technology increase, what used to be long smooth file transfers quickly become short, bursty traffic.

However, if one is trying to achieve, or maintain, an effective sustained data rate on the order of 2.4 Gb/s to 10 Gb/s, as some applications will begin approaching, a large transport protocol window will be needed. In such circumstances, it is very expensive to drop packets and retransmit, while holding gigabits of data in both the message send and receive buffers. This argues for some form of resource reservation along the transmission path, with bandwidth guaranteed until the transmission concludes. Furthermore, this must be done without incurring the latency penalties of a round-trip circuit setup, whether real as in circuit switched networks, or virtual as in transmission control protocol (TCP).

Mills developed the notion of “Just in Time Switching,” in his HIGHBALL project [10]. As he explained, trains leaving Grand Central Station in New York enroute to Los Angeles do not preset the switches across the country. The switch is set just in time as the engine of the train arrives. Mills’s technique used a pre-reservation scheme whereby the desired schedule for a transmission was broadcast to all the switches, and sharing the same scheduling algorithm, all were able to compute the schedule for the request independently, and would know when to dedicate the resource for the scheduled transmission.

McFarland built on Mills’s idea conceptually, and defined JIT signaling, not as a global reservation scheme, but as a modification to the way signaling is done today. One goal was to provide a technique which could simultaneously provide either circuit service or packet service, without necessarily doing packet switching nor circuit switching. A detailed description of a resulting protocol is reported here.

Basically, a setup signaling message would be sent on a dedicated signaling path, which parallels the intended traffic path, and precedes the arrival of the actual traffic on the traffic path by some delta time, without awaiting the full end around delay encountered in a circuit setup. When the transmitter completes transmitting, a completion message would be sent on the signaling path. If the path were held for a short duration of time, the communication resembles that of packet switching. If one has a long holding time, the communication resembles that of circuit switching. There are important advantages in using such a scheme in optical switching network. By separating the signaling control from the data transmission, it permits electronics implementation of the signaling control path, while maintaining a transparent optical data path for high speed data transmission. Another advantage is that it buffers data at the source where electronic memory is cheap and abundant, rather than at the intermediate nodes, where optical delay is expensive and limited.

A few early analyses of the technique have been accomplished. In their Optical Network Control Study [11], Parker*et al.* reported on simulation results on the comparison of circuit switching operation (JITO) and JIT with negative acknowledgment for a metropolitan area size network of optical crossbar switches. The simulation presupposed shortest path routing without contention. Their main conclusion from the simulations was: “The best mode of operation for the all-optical network is the just-in-time with negative acknowledgment. This scheme outperforms circuit switching in all studied scenarios. If the propagation delay is small, this scheme is able to make use of this to get the best performance. If the propagation delay is large, there is a minimum guaranteed performance.”

Hudek *et al.* [12] examined three signaling scenarios through simulation on their TBONE project, an effort that employed identical switches to those considered in [11]. They were: connect/confirmation [CC] (traditional end around circuit setup); tell and go [TAG] (equivalent to our just in time signaling); and just in time switching schemes [RIT]. Their primary conclusion: “When the burst duration to end to end propagation delay ratio is small, all other parameters remaining constant, the TAG NAK scenario shows much improvement over the CC and RIT schemes.”

Wei and Tsai [13] compared JIT signaling with a number of variants of end around circuit setup, and concluded that there were significant efficiencies to be gained by the carriers (efficiency defined as the revenue bearing time of a connection over the total time of the connection), especially on coast to coast calls, using just in time signaling. Their conclusion was based on plotting connection setup delays versus number of hops across the network. A more detailed performance analysis and simulation results of the JIT Signaling scheme has also been published by Wei *et al.* in [24]. They compared the performance of JIT Signaling with those of circuit-switching and packet-switching approaches. Their conclusion: “JIT signaling has the best latency performance among the different switching mechanisms, and it has a better throughput performance than circuit-switching, and its performance is insensitive to network propagation delays.” An analysis of the basic performance of JIT signaling is reported in this paper.

Other variants of optical burst switching schemes have also been examined in the literature. A just-enough-time signaling scheme [14], [15] was proposed by Yoo *et al.*, which attempts to utilize additional knowledge concerning the duration of burst transmission in order to schedule the cross-connect settings in each intermediate switches. Theoretically, due to the reduced channel hold time made possible by forward scheduling, it may deliver better resource utilization than our JIT Signaling scheme. However, in practice it is much more complicated due to additional synchronization and scheduling, and it cannot support discretionary conventional circuit switching.

A terabit burst switching scheme was proposed by Turner [16], and recently an optical label switching scheme was developed by Chang *et al.* [17]. Both schemes are more akin to packet switching in that they require buffering of the packet while its header is being processed. Turner’s scheme was implemented electronically while Chang’s scheme was implemented optically using optical delay lines.

Recently, multiprotocol lambda switching (MPλS) [18] has been proposed as the preferred paradigm for controlling optical cross-connect networks. In MPλS, the wavelength paths are set up using some optical extension of RSVP-TE [19] or CR-LDP [20] protocols. In either protocol, the setup of optical transmission paths incur the same round-trip delay as would in traditional circuit switching signaling. On the other hand, the JIT signaling protocol described in this paper can also be adapted to the MPλS framework to enable the fast dynamic circuit setup needed to support the next generation of optical networking applications.

III. WDM SWITCHING PARADIGMS

In optical WDM, the tremendous bandwidth of a fiber (potentially a few tens of terabits per second) are demultiplexed into many independent nonoverlapping wavelength channels. In an all-optical implementation, within certain restrictions, the wavelength channels are *transparent* in that they can transport data at different bit rates and modulation formats.

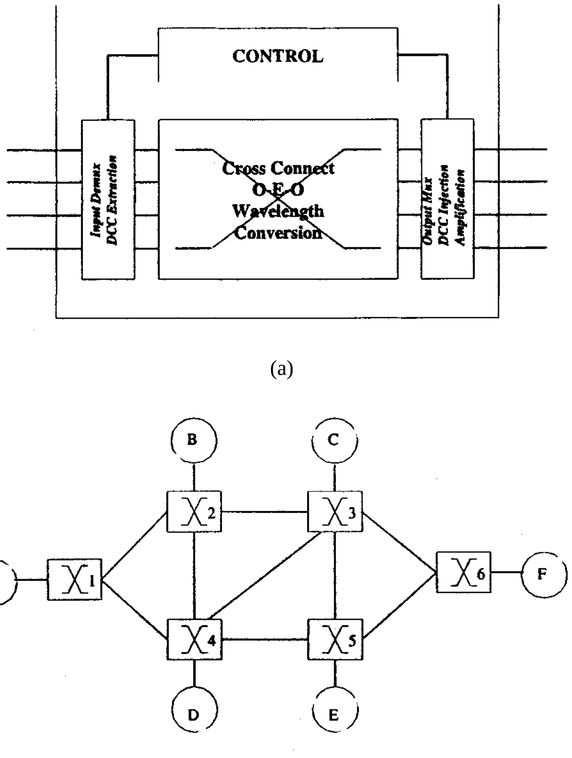
In this section, we outline a functional architecture for a generic WDM switch used for our subsequent discussion. We then describe the different possible switching paradigms for optical WDM in more detail. We note that each switching paradigm makes different assumptions on the WDM switch hardware, and requires different signaling schemes. The manner in which the header/control information is exchanged and the manner in which the path-setup and data-transfer are performed distinguish the different schemes.

A. WDM Switch Functional Architecture

A functional architecture of a generic WDM switch is illustrated in Fig. 1(a). Wavelength channels in an input fiber that enter a WDM switch are demultiplexed into individual wavelengths. They are then switched by the cross-connect to a specific output port. All wavelength channels destined to an output port are then multiplexed into the output fiber. The wavelength channels may be amplified and/or gain stabilized before they exit the switch. WDM switches [21], [22] differ in the extent to which they keep the optical signals transparent within the switch. In an all-optical WDM switch, the wavelength channels remain entirely in the optical domain. Such WDM switches may also perform wavelength conversion within the switch. Limited input and/or output buffering may also be provided in an optical switch via the use of optical delay lines as optical buffering technologies are still immature at this stage. Other WDM switches have optical-to-electronic (O/E), and electronic-to-optical (E/O) conversions performed on each wavelength channel within the WDM switch. Compared to their all-optical counterparts, these electro-optic switches are much less transparent to the data signal formats (i.e., protocols) and bit rates. Depending on the control signaling schemes used, extraction and injection of separate data communication channels (DCC) may also be included [23].

The control component in a WDM switch controls the state of the cross-connect, the wavelength converters, and input–output buffers, based on the control information present at the controller. An important parameter of a WDM switch is the *switching-time*, defined as the time it takes for the cross-connect to change state, and for the output channel to stabilize.

Fig. 1(b) depicts an optical network consisting of WDM switches (labeled 1 through 6) interconnected by bidirectional fiber links. Technological constraints dictate that the number of WDM channels that can be dynamically switched in an optical cross-connect is limited to W, whose value is typically up to 64 today, but are soon expected to grow to a few hundred. Access stations are attached to a WDM switch via bidirectional fiber links. An access station is assumed to be capable of sourcing data on any one of the available wavelengths, e.g., by using tunable lasers. Data is transferred between access stations as unidirectional variable length optical bursts.



(a)

(b)

Fig. 1. Multiwavelength optical networking architecture.

There are different ways to transmit the control information across the WDM network nodes. Control information can be transmitted along with the optical data as an in-band optical header (e.g., by utilizing a scheme such as subcarrier multiplexing). In this mode, control information travels along with the data-burst, and is analogous to packet switching. An implication is that the WDM switch must delay/buffer the data-burst until the header is decoded and the data-path is established. An alternative is to transmit control information independent of the data-burst on a separate signaling channel. In this case, the control information (i.e., signaling) is utilized to setup the optical path prior to transmitting the data-burst, and is analogous to circuit-switching. Intuitively, a packet-switched scheme is suited for short data-bursts, and a circuit-switched scheme is efficient for long data-bursts. (The notion of “long” with respect to what is discussed below).

Current optical technology provides very high bandwidth for transmission, but is limited in its ability to perform optical processing or buffering. Electronics on the other hand allows processing and buffering, but cannot match the bandwidth of optical transmission. In order to maximize the utilization of the

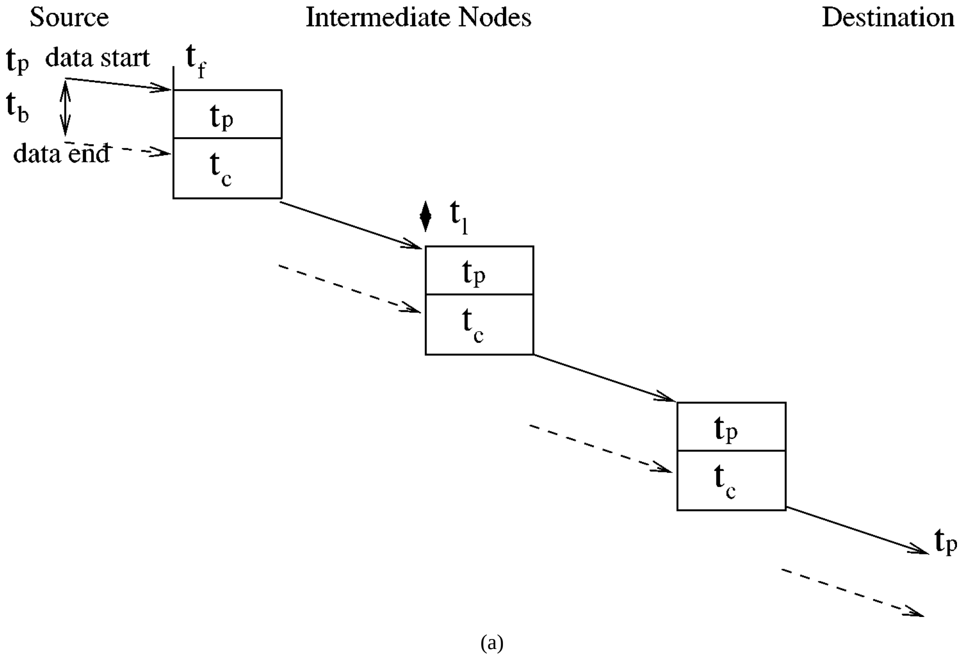


Fig. 2. Conventional switching paradigms. (a) Packet switching.

optical network, we would like the control mechanism to minimize the time it takes to setup the optical path for a data-burst, under the constraints imposed by the optical WDM switch technology. JIT optical burst switching is designed to combine the desirable features of packet-switching (small setup time) and circuit-switching (out-of-band signaling).

B. Packet-Switching

In optical packet-switching [22] as illustrated in Fig. 2(a), the control information associated with a data-burst travels with the data-burst as the packet header. At each intermediate node, the header is separated from the data-burst, and is processed to determine the output-port. A routing protocol may be used to provide the routing table for the next-hop given the destination. The WDM switch controller sets up the cross-connect (along with wavelength conversion if any). During the period of header processing and cross-connect setup, the data-burst is temporarily buffered. Due to the very limited optical buffering capability, if an output port is not available, the data-burst is typically dropped. No explicit feedback concerning the fate of the packet is sent to the source access station at this level.

C. Circuit-Switching

In optical circuit-switching, the data transfer path is setup prior to the transmission of the data burst. The source access station initiates an out-of-band distributed signaling procedure to determine the path, wavelengths, and setup cross-connects. When a data transfer path is available, the data-burst is transmitted by the access station.

An example of the circuit-switched signaling procedure is depicted in Fig. 2(b). A SETUP message is sent from source to destination. On its way, wavelengths are reserved at each link along the path. If at some intermediate node no wavelengths are available on the output port, then a BLOCKED message is sent back to the source. If the setup message reaches the destination access station successfully, then the destination responds with a CONFIRM message back to the source along the reverse path, and at each intermediate node, a cross-connect is setup. When the confirm message reaches the source, the source transmits the data-burst. After the data-burst is transmitted, the source sends a RELEASE message which releases wavelengths along the path. We assume that the routing is performed by a routing algorithm and provided to the switch by a routing protocol, both of which are independent of the signaling protocol.

The setup time for a data-burst in circuit switching can be improved by pipelining the cross-connect setup times with the propagation time. Two variations of pipelining are possible:

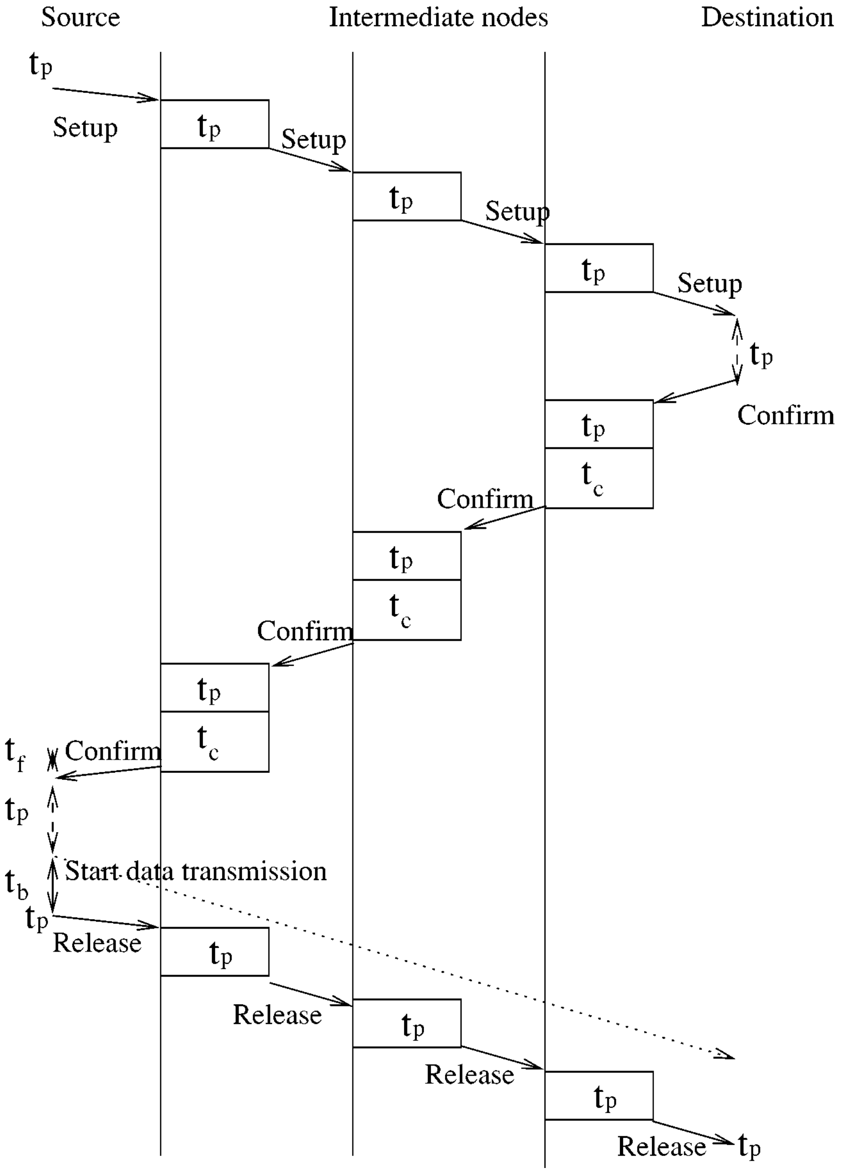
- *Cut-at-Confirm:* where the cross-connect is configured (cut-through) after the CONFIRM message is sent [see Fig. 3(a)] on the reverse path.
- *Cut-at-Setup:* where the cut-through is performed right after the SETUP message is sent [see Fig. 3(b)] on the forward path.

In either case, actual reservation of the crossconnect channel (as oppose to crossconnect channel cut-through) is done as part of the SETUP message processing.

D. Just-In-Time Optical Burst Switching

We observe that in circuit switching, regardless of the specific pipeline scheme, a round-trip delay is always incurred in the setup time. This translates into a significant setup up delay as the network diameter increases. Furthermore, the intermediate switches are all locked into the desired cross-connect states well before the actual arrival of the data burst. This leads to inefficient usage of cross-connect bandwidth. Therefore circuit switching is efficient only for data-bursts which are much longer than the setup time.

The packet switched scheme on the other hand has shorter setup times since the control information travels with the data burst. However in an all-optical WDM switch, the header



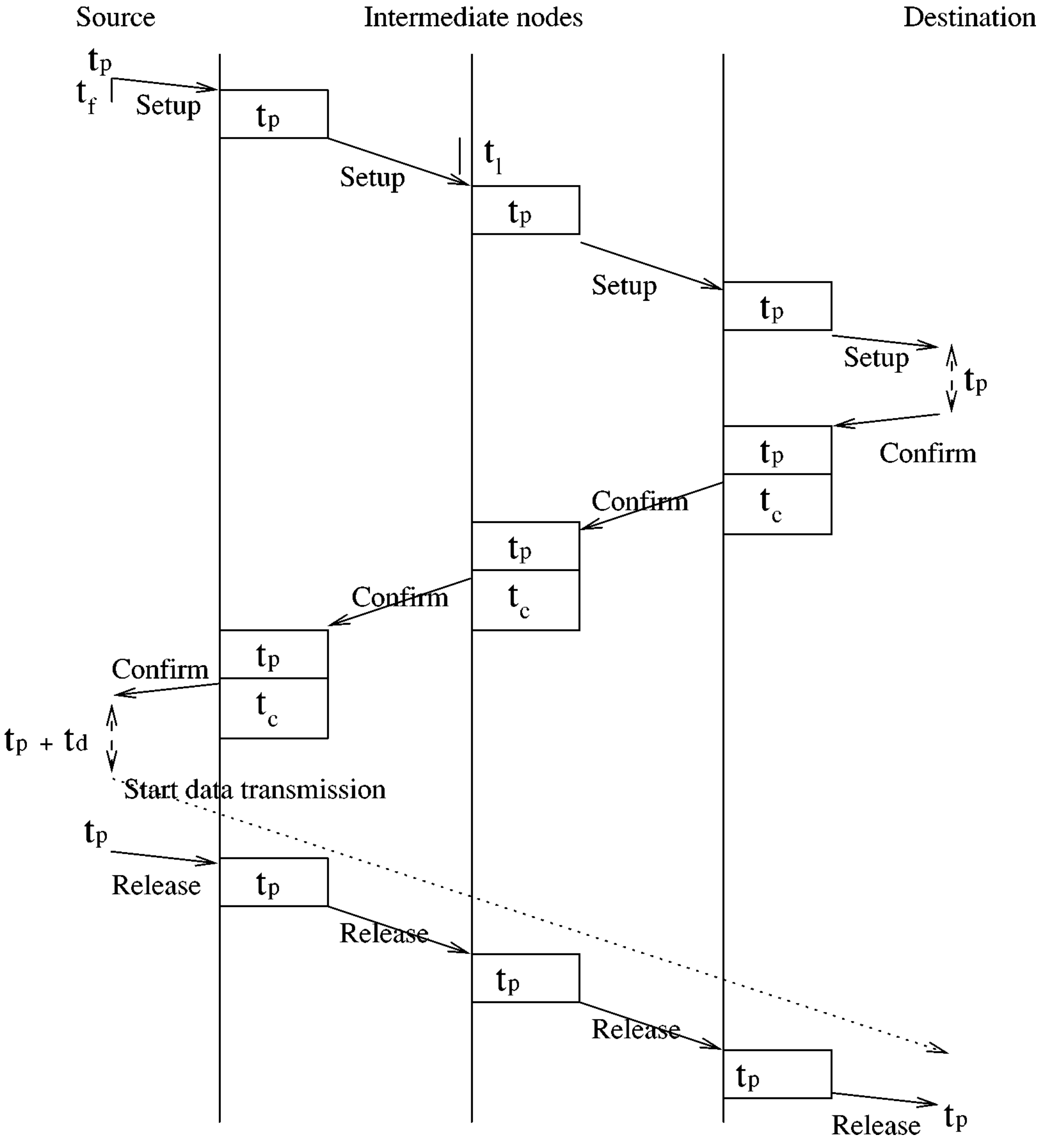
(b)

Fig. 2. *(Continued.)* Conventional switching paradigms. (b) Circuit switching.

processing requirements can be significant. For example, a WDM packet switch must have the technology to extract the header from the optical data-burst at the input port and to reimpose it at the output port. Such technology is not required for the circuit-switched schemes. Furthermore in packet switching, the data burst needs buffering at every node during the time that the packet header is being processed. Depending on the switching-time, this may require significant size of delay lines, as this requirement is imposed on a per wavelength channel basis.

Just-in-time optical burst switching combines desirable features of circuit-switching with those of packet-switching. It features out-of-band control signal processing that eliminates buffering of data-burst at intermediate nodes, while minimizing the setup time, and maximizing the cross-connect bandwidth efficiency.

Fig. 4 depicts a simplified abstract form of signaling for JIT-OBS that captures the salient features for our present discussion. A more detailed description will be presented in Section V. An access station initiates JIT signaling by sending a SETUP message to its attached WDM switch. The WDM switch responds with a CALL_PROCEEDING to indicate that connection setup is on its way to the destination. In the reply message there is a *delay* parameter, that indicates how long the access station should wait before launching its data-burst. This delay parameter is estimated by the WDM switch (e.g., by a suitably developed routing algorithm) from the number of hops to the destination and associated setup time of the cross-connects along the path. When the source access station receives the CALL_PROCEEDING message, it waits for the specified delay duration and then transmits its data-burst. When a WDM switch receives a SETUP message, it will attempt to reserve the wavelength on the output port and forward the SETUP message to its next hop. Cross-connect setup is performed in parallel with the next hop propagation. When the destination access station receives the SETUP message, it



(a)

Fig. 3. Pipelined circuit switching paradigms. (a) Cut-at-confirm.

responds with a CONNECT message which travels back to the source access station. After transmitting the data-burst, the source access station transmits a RELEASE to release all reserved wavelengths.

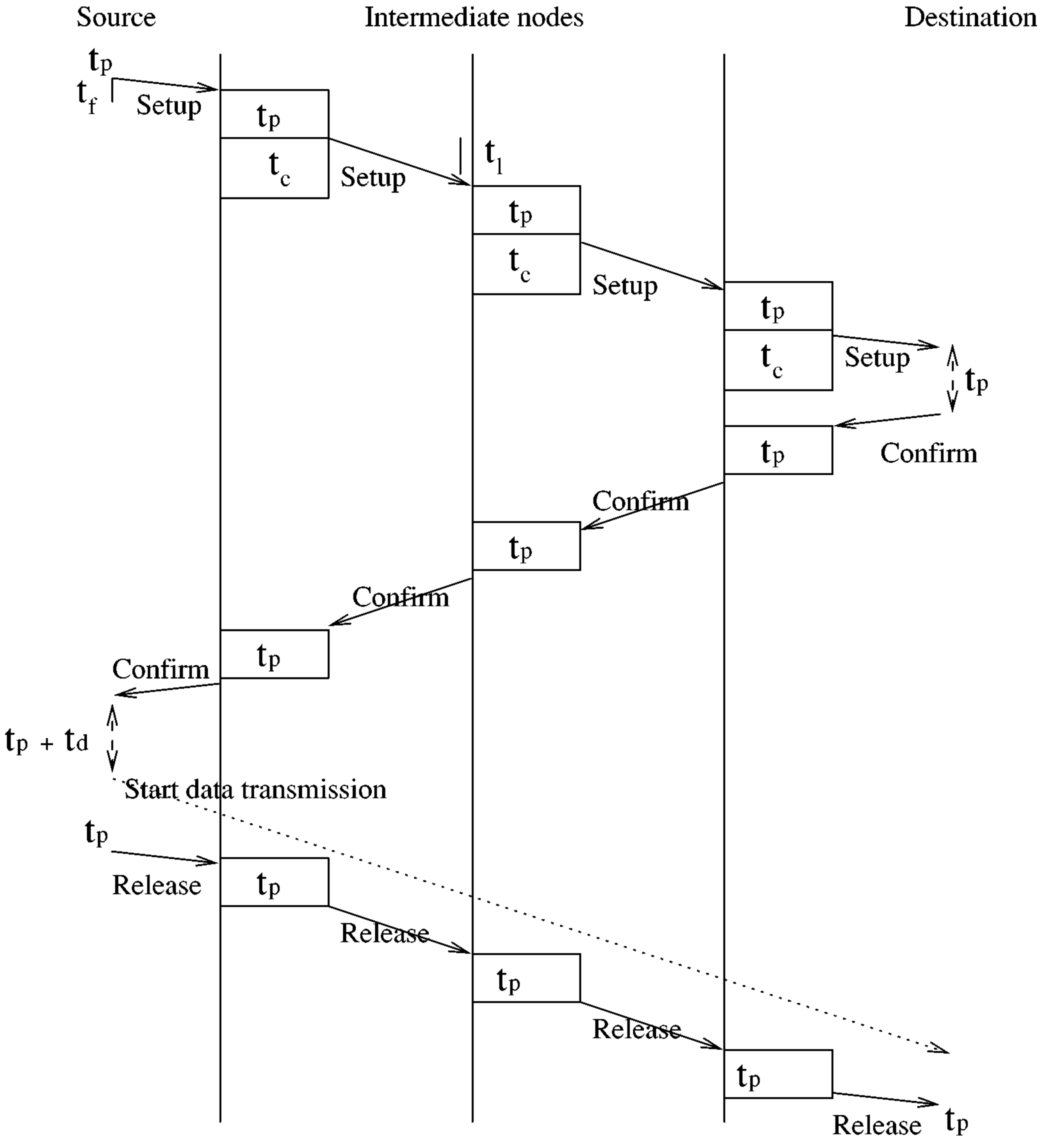
Observe that in JIT-OBS, the data-burst is buffered at the source where electronic memory is cheap and abundant, rather than at the intermediate switching nodes where optical delay is expensive and limited. JIT signaling also provide a best-effort explicit acknowledgment of end-to-end path setup which allows it to be used to emulate conventional circuit switching signaling at the sender's discretion.

E. JIT Signaling Performance

In this section, we present an analysis of the basic performance of the JIT OBS scheme. We contrast it with those of the conventional switching schemes. A more detailed performance analysis can be found in [24]. In analyzing the performance of the different switching schemes, we are interested in the source setup time and the channel hold time values. The source setup time affects the overall end-to-end latency of the data-burst transmission, as well as the potential source buffer requirements. The channel hold time affects the network's resource usage efficiency in supporting dynamic bursty traffic.

1) Assumptions and Notation: We assume that the propagation delay on each network link is identical (i.e., the fiber links are of the same lengths). Likewise, we assume the propagation delay on each access link is identical. We also assume that the processing delays of the signaling messages (setup, release and call proceeding) at all nodes (source, intermediate switch, or destination) are identical. For each signaling scheme, we derive the formula for the source setup time, and the channel hold time (i.e., the time duration from the instant a channel is reserved to the instant the channel is released) for a successful data-burst transmission. We now define the notation utilized in the following analysis:

t_f: propagation delay from an access station to its attached WDM switch;



(b)

Fig. 3. *(Continued.)* Pipelined circuit switching paradigms. (b) Cut-at-setup.

t_p: protocol message processing time at each node (intermediate WDM switches, source, and destination);
t_c: crossconnect cut-through switching and stabilization time at a WDM switch;
t_l: propagation delay on a fiber link between WDM switches;
t_b: data-burst transmission duration.

These parameters are illustrated in Figs. 2–4 for each of the different switching schemes. Typical values for these parameters for current generation of WDM switches are: fiber delay of about 5 μs per kilometer, signaling protocol processing time of a few μs, and crossconnect cut-through time of a few ms.

2) Source Setup Time: The *setup time* for a data-burst transmission is defined as the *duration from the instant the data-burst presents itself at the source for transmission, to the instant that such a transmission commences*. For a route with n intermediate WDM switches ($n+1$ hops), the setup time for the different schemes can be derived by summing the different delay components depicted in Figs. 2–4 that are incurred before transmission can commence.

For *Packet Switching:* the source setup time is simply the source setup processing delay: t_p.

For *Circuit Switching*, the source setup time is given by the whole round-trip time:

$$4t_f + 2(n-1)t_l + (2n+3)t_p + nt_c.$$

For *Cut-at-Confirm Circuit Switching*, the source setup time is

$$4t_f + 2(n-1)t_l + (2n+3)t_p + t_d$$

where $t_d \geq t_c - t_p - 2t_f$, $t_d \geq 0$, is an additional source setup delay required to assure cut-through completion at the first switch.

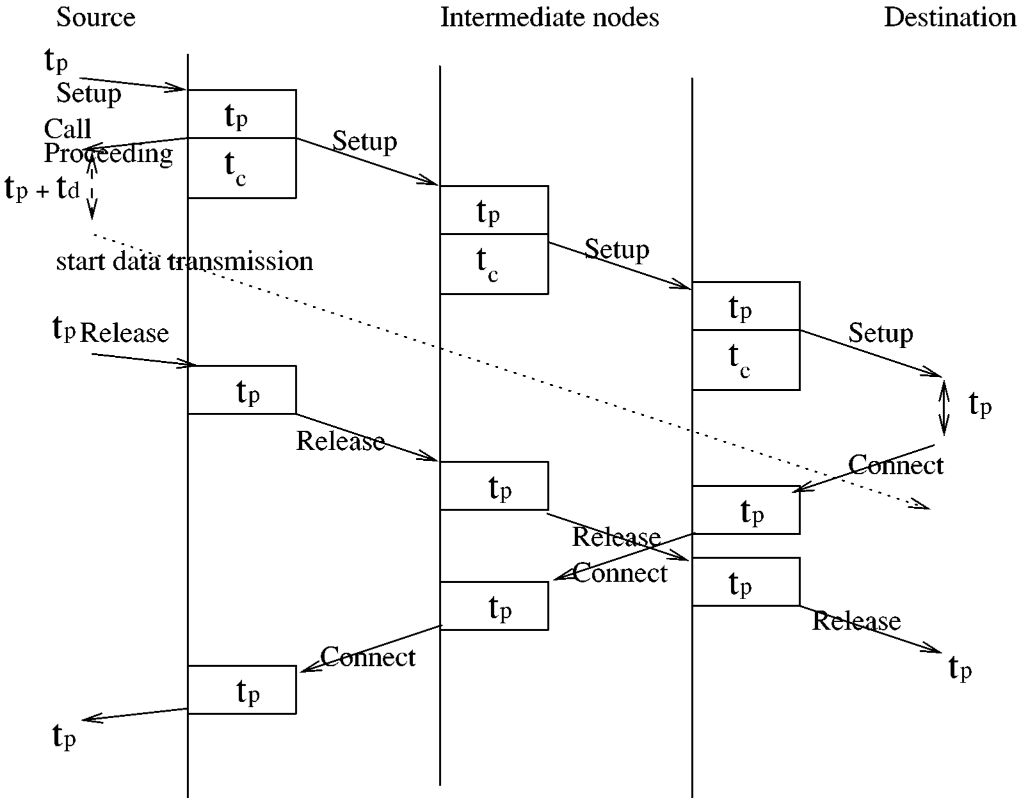


Fig. 4. Just-in-time optical burst switching.

Likewise, for *Cut-at-Setup Circuit Switching,* the source setup time is

$$4t_f + 2(n-1)t_l + (2n+3)t_p + t_d$$

where $t_d \geq t_c - (4t_f + 2(n-1)t_l + (n+3)t_p)$, $t_d \geq 0$, is the additional source setup delay required to ensure cut-through completion at the last switch.

For *JIT Optical Burst Switching,* the source setup time that is required before actual data-burst transmission, is determined by the requirement that, for any switch along the path, its cut-through must complete before the data-burst arrives at the switch. In addition, the destination must also have completed its signaling processing before the data-burst arrives. Therefore, the required source setup time is given by

$$(n+2)t_p + t_\delta$$

where $t_\delta \geq t_c - t_p$, $t_\delta \geq 0$, is the extra delay adjustment required to assure cut-through completion at the last WDM switch. Alternatively, let t_d be the additional source setup delay to be imposed upon receipt and processing of the JIT call proceeding message before data-burst transmission commences, then we have

$$t_d = (n-1)t_p - 2t_f + t_\delta.$$

We observe that JIT-OBS has significantly lower source setup time than those of the circuit switching schemes.

3) End-to-End Latency: The *end-to-end latency* for a data-burst is defined as the *duration from the instant the data-burst presents itself for transmission at the source, to the instant it is completely accepted at the destination*. For a route with n intermediate WDM switches ($n+1$ hops), the end-to-end latency for the different schemes are summarized as follows.

- *Packet Switching:* $2t_f + (n-1)t_l + (n+2)t_p + nt_c + t_b$.
- *Circuit Switching:* $6t_f + 3(n-1)t_l + (2n+3)t_p + nt_c + t_b$.
- *Cut-at-Confirm Circuit Switching:* $6t_f + 3(n-1)t_l + (2n+3)t_p + t_b + t_d$.
- *Cut-at-Setup Circuit Switching:* $6t_f + 3(n-1)t_l + (2n+3)t_p + t_b + t_d$.
- *JIT Optical Burst Switching:* $2t_f + (n-1)t_l + (n+2)t_p + t_b + t_\delta$.

These equations are again derived by summing up the different delay components in the end-to-end switching scenarios depicted in Figs. 2–4. The t_d and t_δ parameters for the different schemes are the same as their respective values in the source setup time formulas in Section III-E2. Again we observe that JIT-OBS has the lowest latency among all schemes.

4) Channel Hold Time: We define the *channel reserve time* to be the instant at which an outgoing channel is reserved at a WDM switch for a call. *Channel release time* is the instant at which an outgoing channel that was reserved for a data-burst is subsequently released. *Channel hold time* is then defined as the *duration for which an outgoing channel is reserved by a WDM switch for a data-burst*. Let n be the number of intermediate WDM switches along a path. Let time $= 0$, when a data burst presents itself for transmission at the source. From Figs. 2–4, we make the following observations.

For the packet switching scheme the channel hold time at any WDM switch is simply the sum of the cut-through time, the burst duration, and the release processing time

$$t_c + t_b + t_p.$$

For the circuit-switched scheme, a setup request processing is completed at the kth WDM switch on its path at time $t_f + (k-1)t_l + (k+1)t_p$. This is when an outgoing channel is reserved at the kth node. After the data-burst has been transmitted, the output channel can be released at the kth switch at time $5t_f + (2n-3+k)t_l + (2n+4+k)t_p + nt_c + t_b$. Therefore the duration for which the channel is reserved for a call, i.e., the channel hold time, at the kth switch is

$$4t_f + 2(n-1)t_l + (2n+3)t_p + nt_c + t_b.$$

Observe that the channel hold time is equal to the corresponding source setup time plus the data-burst transmission duration, and it is identical across all WDM switches. This is because the setup message and the release message both incur *identical* propagation and processing delays along the same route under our assumptions. In fact, this property is shared by all three circuit-switching schemes as well as by the JIT-OBS scheme.

For the cut-at-confirm circuit-switching scheme, the channel hold time at a WDM switch is therefore simply

$$4t_f + 2(n-1)t_l + (2n+3)t_p + t_d + t_b.$$

where $t_d \geq t_c - t_p - 2t_f$, $t_d \geq 0$, as determined above in Section III-E2.

Similarly, for the cut-at-setup circuit-switching scheme, the channel hold time at a WDM switch is simply

$$4t_f + 2(n-1)t_l + (2n+3)t_p + t_d + t_b$$

where $t_d \geq t_c - (4t_f + 2(n-1)t_l + (n+2)t_p)$, $t_d \geq 0$, as determined above in Section III-E2.

Likewise, for the JIT-OBS scheme, the channel hold time is given by

$$(n+2)t_p + t_\delta + t_b$$

where $t_\delta \geq t_c - t_p$, $t_\delta \geq 0$, as determined in Section III-E2. Assuming a nonzero tight bound on the value of t_δ, then the channel hold time can be expressed as

$$(n+1)t_p + t_c + t_b$$

Observe that while the channel hold time for the JIT-OBS scheme is larger than that of the packet switching scheme, it is independent of the link propagation delay, and is smaller than any of the circuit switching schemes. The savings are especially prominent in long distance transmissions. This translates into significant improvement in network efficiency and throughput. Furthermore, practical implementation of JIT-OBS is likely to be far simpler than optical packet switching.

IV. DESIGN ASSUMPTIONS AND OBJECTIVES

In this section, we enumerate and discuss the design assumptions and objectives in our development of JIT signaling protocol for WDM optical burst switching. JIT Signaling employs an out-of-band signaling scheme. We assume that the WDM optical networks provides a separate signaling channel (which is referred to as the data communication network (DCN) in MONET, and is implemented using a raw ATM channel) on which to carry the JIT signaling protocol messages. The DCN provides bidirectional communications between neighboring network elements. This enables bidirectional control handshake messages. While we recommend that the signaling channel be co-resident on the fiber carrying the traffic, our design does not require it.

The JIT signaling function requires access to a proper routing function and a wavelength assignment function. The routing function and the wavelength assignment function should be integrated in the same protocol for efficient JIT call setup. The subject of routing and wavelength assignment in JIT signaling is beyond the scope of the present paper and is reserved for future study. Regardless, the routing and the wavelength assignment functions are external to JIT signaling, and we have tried to place as few restrictions upon these functions as possible. We do stipulate that the routing and wavelength assignment protocol, when given a destination address within the WDM network address plan, will return at a minimum, the next hop toward the destination and a wavelength to be used for the data-burst transmission toward the next hop.

In JIT signaling, we assume that the client equipment is capable of sourcing data on one or more of the available wavelengths, e.g., by using tunable lasers. To transmit, a signaling client will request an initial preferred input wavelength to the edge switch, e.g., the wavelength that a client's laser is already tuned to. Depending on the switch hardware capability (e.g., wavelength conversion,) and the ensuing wavelength utilizations in the network, the wavelength selection function at an edge switch may or may not choose to honor such a request. We note that client equipment that is able to support transmission on any available wavelength, provides maximum flexibility to the wavelength assignment function of the network.

Because of the just in time nature of the signaling messages, it is necessary to examine the desired level of message error handling functions to be provided. Standard protocol message error handling involves detection of corrupted messages, and may include retransmission of lost or corrupted messages. In JIT-SP, the retransmission function is not useful for all signaling messages. This is because a retransmitted call setup message is likely to arrive too late at the downstream intermediate switches and misses the deadline required for successful just-in-time data transmission. Similar reasoning argues that supporting crankbacks[1] in call setup path hunting is not likely to be productive. The situation for call release messages is more complicated. We have contradicting concerns here.

- On one hand, retransmission may be justified for the call release messages since they are not time critical, and their loss (or corruption) could result in unreleased resources and degrade network performance.
- On the other hand, we expect the typical error rate of signaling message transmission in optical networks to be so low that a complicated retransmission may not be justified. The very rare cases of corrupted messages are deemed acceptable, as long as we can detect the corrupted message based on a simple and efficient scheme such as a cyclic redundancy check (CRC) scheme [25].

To resolve this, in our design, we require retransmission capability to be provided for the call release messages. But we also stipulate that retransmission can be enabled/disabled via net-

---

[1] Call setup rollback and alternative path retry.

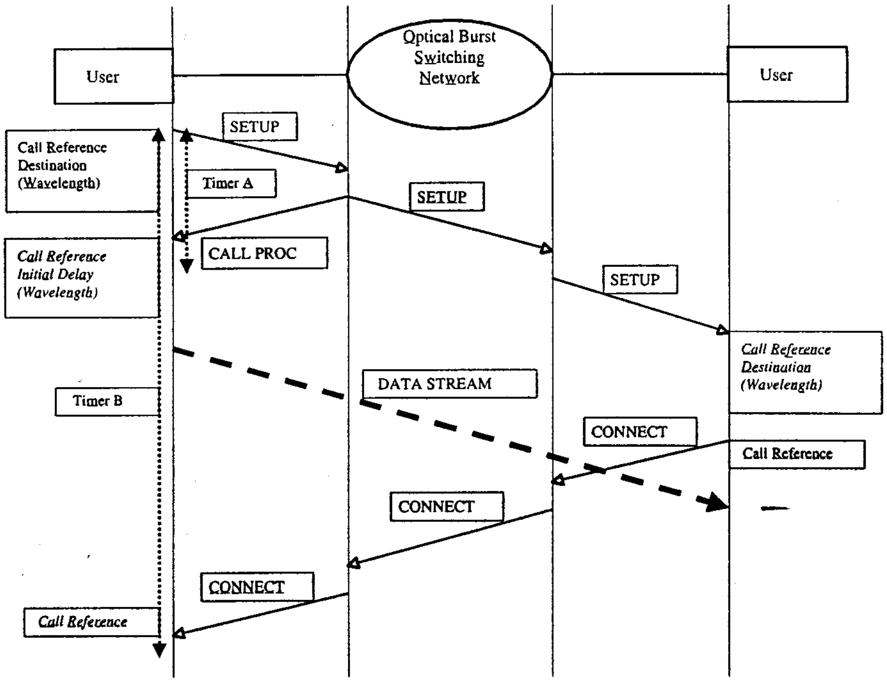


Fig. 5. Successful just-in-time connection establishment.

work management so it can be tailored based on operational experiences.

In summary, the MONET JIT signaling protocol for WDM optical burst switching is designed with the following objectives.

- The protocol shall support unidirectional point-to-point data transmission. It provides the signaling clients the capability to originate and receive a point-to-point optical data burst dynamically.
- The protocol shall support best effort dynamic setup for the transmission of data. Possible retransmission of a databurst if so desired, is deemed the client application or a higher level protocol's responsibility.
- The protocol shall support both packet services as well as conventional circuit-switched services. The exact mode of operation shall be the sender's discretion, but one JIT implementation is capable of providing both depending on how it is used.
- The protocol shall segregate the concerns of routing and wavelength assignment from signaling. This way, different routing approaches can be employed and experimented under the same signaling design.

In the following sections, we show in detail what the signaling protocol is like when using the JIT approach.

V. SIGNALING SCENARIOS

In this section, we describe several scenarios that detail the message dialogs between the signaling clients and the WDM switches as they employ the JIT signaling protocol to establish and tear down connections.

Depending on the ensuing network condition, a JIT connection attempt may result in different outcomes. The JIT connection request may succeed end-to-end, or it may fail due to connection request rejection or signaling message lost. A common reason for rejecting a JIT setup request is that the intermediate switches cannot satisfy the request because there is no acceptable outgoing channel resource (routing tables may support mutliple possible output channels to complete setup) available at the time, i.e., a collision. Rejection of the setup request may also be due to other reasons such as destination unknown, or signal incompatibility etc.

A. Successful Connection Establishment

Fig. 5 illustrates the case for a successful JIT connection establishment. In this case, an end-to-end connection path is successfully set up. Notice that the data stream arrives at the destination user node after it is ready to accept the stream by having indicated CONNECT back to the network.

The source sends a SETUP message to the WDM network element to which it is attached. The SETUP message carries with it a number of message information fields. In addition to the destination (called party) address, a *call reference* is used to identify the new call uniquely in subsequent communications between the network and the user. An optional wavelength identifier can be specified to request setup on a specific wavelength.

Upon receiving a SETUP message at the first NE, the JIT signaling agent consults a routing function and responds with a CALL_PROCEEDING message. Among other things, this message contains two important pieces of information for the originating user node:

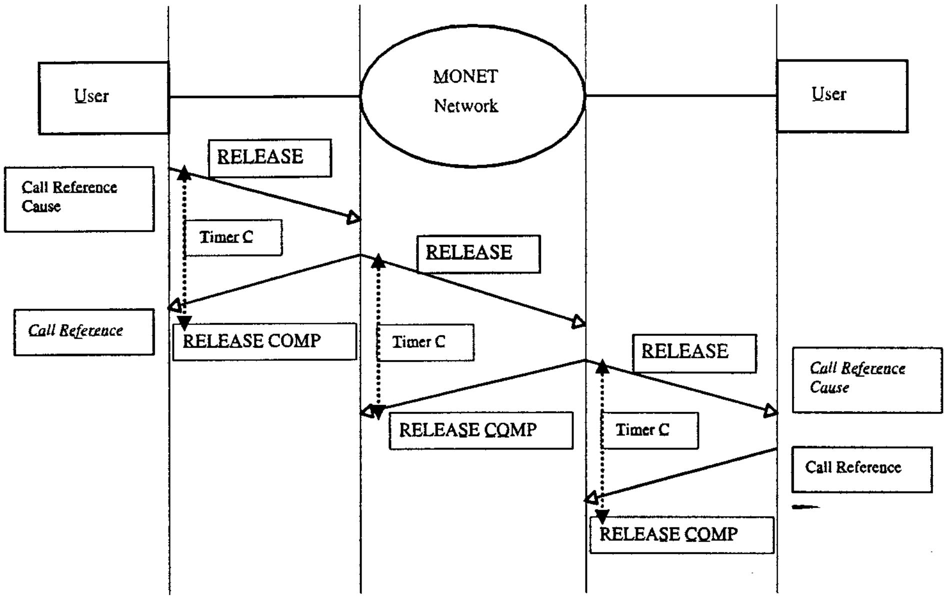


Fig. 6. JIT connection tear down.

- *Estimated Initial Delay*—which advises the originator how long it should wait before sending out the data stream for JIT transmission.
- *Output Wavelength*—which tells the originator which wavelength it should use to output its data transmission.

The delay parameter is estimated by the first signaling agent from its estimate of the number of hops to the destination and associated setup time of the cross-connects along the path. When the source access station receives the CALL_PROCEEDING message, it waits for a duration suggested by the estimated delay parameter before it starts data transmission. Meanwhile the WDM switch reserves the wavelength on the output port, proceeds to make the actual cross-connect by issuing a command to the fabric controller, and forward the SETUP message to the next hop. It also stores this cross-connect information along with the call reference for later retrieval. The SETUP message travels hop-by-hop each time consulting the local routing table and resource availability, making cross-connects, until it reaches the destination. When the sink station receives the SETUP message, it responds with a CONNECT message which is sent back to the source. Meanwhile, the data-burst may have been sent and it will hopefully reach the NEs along the trail just in time after a cross-connect has been made at each point along the trail. In the case where a short data-burst is being transmitted over long distance, it is possible for the data transmission to have ended and the connection released before the CONNECT message reaches the sender. In such case, the CONNECT message will be quietly dropped.

Observe that since the initial delay parameter is only an estimate, having a successful end-to-end trail setup (as indicated by receipt of the CONNECT message at the source end) is not enough to guarantee successful transmission of the data-burst. A higher level protocol will be needed to guarantee reliable end-to-end transmission. On the other hand, the source access station may elect to withhold data transmission until it receives the CONNECT message issued by the destination station. Doing so will guarantee that a conventional end-to-end circuit is in place before transmission commence. Therefore, a source station can select either a JIT switching mode or a conventional circuit-switching mode to transmit its data.

Timers are needed at the source to make sure that the messages and actions are carried out within specific time constraints to support burst switching. Timer A is set to the maximum amount of time the client has to wait for the CALL_PROCEEDING message to arrive. A time-out on timer A indicates a call-setup failure. Timer B, on the other hand, is the maximum allowable waiting time for a client to receive a CONNECT message indicating a successful end-to-end trail creation. A time-out on timer B may indicate a failure in the end-to-end setup. If either timer expires, the client aborts the transmission request. It is the client's decision to set the timer and determine what the allowable time constraints are according to its applications.

B. Connection Tear-Down

Fig. 6 illustrates the sequence of messages exchanged between the network and the originating and destination users in a typical connection teardown scenario. After successfully transmitting a data burst, the source access station transmits a RELEASE message with a call-reference parameter to identify which cross-connect to release along the path. The RELEASE message will indicate a normal termination of connection in its cause field. Each NE in the connection trail responds with a RELEASE_COMPLETE to indicate a successful release of connection.

The tear-down process in fact can be initiated by either the originator or the destination. When a RELEASE is sent from

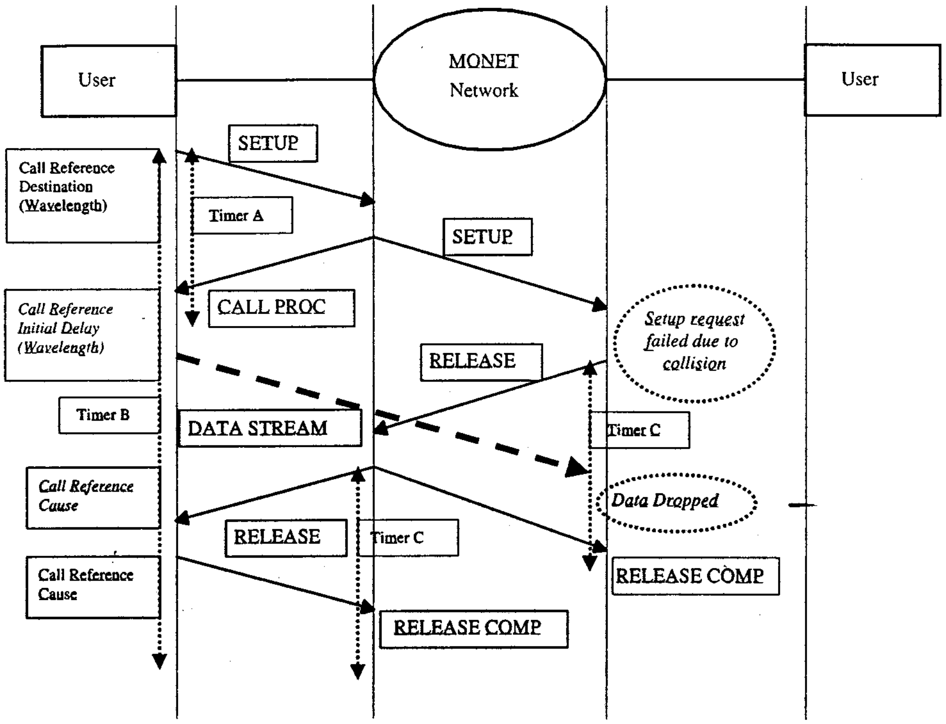


Fig. 7. Failed JIT connection due to rejection.

the source, the signaling agent at the intermediate NE's checks for the call reference. If it finds a cross-connect associated with the call reference, it issues a disconnect to its NE controller. At the same time, it sends an acknowledgment, i.e., a RELEASE_COMPLETE, to the immediate source of the RELEASE message. Then it will forward the release request to the next hop. The next hop will do the same and the cycle goes on until the RELEASE message reaches the other end of the trail completing the end-to-end release of the connection.

Observe that the RELEASE message cannot be processed if the signaling agent at an intermediate switch crashes. The RELEASE message retransmission will abort after a specified number of times. In this case, unreleased cross-connects will remain in the network. A separate cleanup algorithm may be implemented and be invoked by network management to handle such cases.

C. Connection Failure due to Request Rejection

Fig. 7 illustrates the case for a failed JIT connection establishment attempt. Again, as before, the SETUP message is initiated at the source. Then a CALL_PROCEEDING message is sent back to confirm the SETUP request and client waits for an estimated delay suggested by the first network element agent before it starts sending the data. But in this case, as the message travels down-stream and the cross-connects are being made, a collision could occur along the intended path. The cause of conflict could be unavailable wavelength, port, route, etc. This event triggers a call clearing to propagate back (recall that we do not support crank-backs in path hunting). A RELEASE message is sent back to the source. The exact cause of the release is encoded and sent back to the source along with the release message. To guard against (unlikely) loss of the RELEASE message which may result in unreleased resources, a RELEASE message is explicitly acknowledged by a corresponding RELEASE_COMPLETE message, and a timer (Timer C) is used to trigger retransmissions if necessary.

When the RELEASE message reaches the source, the client has the option to either retry the connection, or to abort altogether. This decision will probably depend on whether the condition causing the release is transient or likely to be persistent. This information is also embedded in the cause field. Normally this release message should have been received by the source before timer B times out so that it would not issue a release of its own. However, if a short data-burst is being transmitted over long distance, it is possible for the data transmission to have ended before the collision generated release message is relayed back to the sender. Meanwhile, the sender may have issued its own release following the completed transmission and considered the transmission a success at this layer.[2] This discussion clearly highlights the best-effort nature of the JIT switching paradigm. A higher level protocol is needed to guarantee reliable end-to-end transmission.

D. Connection Failure due to Message Lost

Fig. 8 shows another scenario with a failed JIT connection attempt, but this time, it is caused by the signaling message getting dropped or lost. This may be caused by an unreliable

---

[2] If short-burst and long-distance transmissions are frequent, it may be justified from a performance perspective to implement automatic retransmissions at the client protocol stack, thereby emulating a "reliable" JIT datagram service.

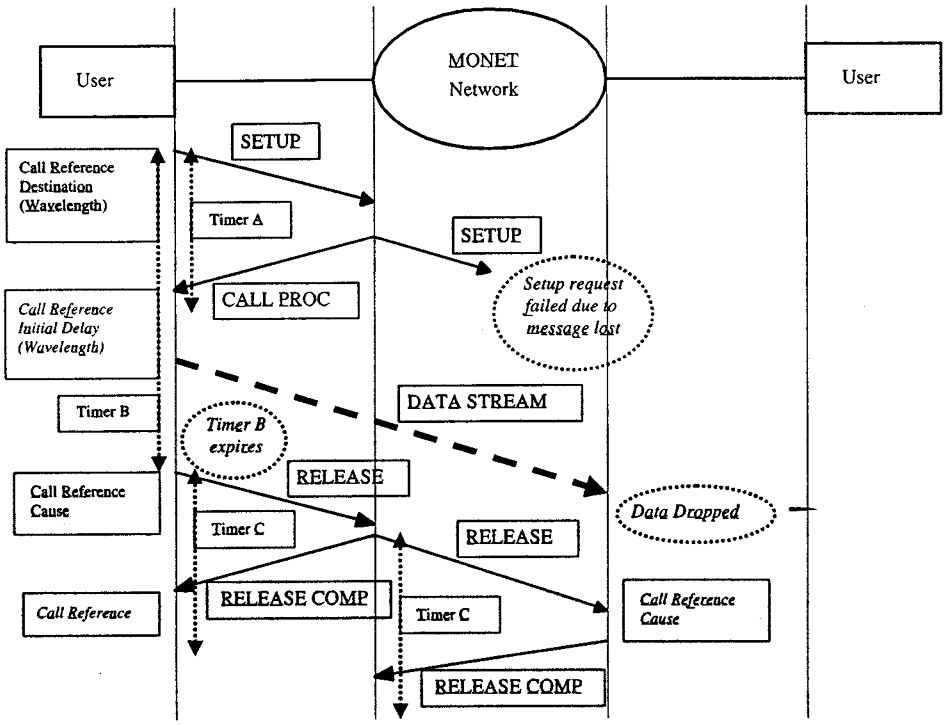


Fig. 8. Failed JIT connection due to message loss.

data communication channel, or that one of the signaling agent has crashed or is unable to respond in some way. In this case, the source sends a SETUP request, the first NE responds with CALL_PROCEEDING as before, makes the cross-connect, and forward the SETUP message to the intermediate hops. This time the message gets dropped along the way. Neither the last NE nor its next hop NE knows what happened to the SETUP message because there is no acknowledgment of setup requests at the intermediate nodes. But eventually, the originating client detects a timeout on timer B and sends a RELEASE message using the same call reference. The RELEASE message propagates down from the originator to the point where the message was lost, hopefully releasing all cross-connects made in the partial trail. Meanwhile, any data that have been transmitted are dropped at that point.

Again there is no automatic recovery support from the signaling protocol, and it is up to the source to determine whether to retry the connection or to abort. Furthermore in this case, there is no "cause" feedback to provide a clue as to what happened in the setup request or even where it failed. The client will have to decide how many times it will retry depending on its application requirements. In a similar situation, when a corrupt message is detected at any point, the message will be dropped. In such situation the scenario described above also applies. Message validation can be implemented by using a simple cyclic redundancy check [25].

VI. JIT SIGNALING MESSAGES AND INFORMATION FIELD

In designing the set of MONET JIT signaling messages, we are motivated by the following considerations:

- Keep the layout and format of the message and its information fields as fixed and as regular as possible to simplify parsing and permit hardware interpretation of the routine configuration change messages for minimum delay.
- Since MONET uses ATM as its DCN, the messages should all fit in a single ATM cell. This will allow us to pass the messages directly over ATM AAL0 (raw ATM.)

Doing so makes possible high-performance implementation that is simple and efficient.[3]

A. Signaling Messages

The MONET JIT Signaling Protocol defines the following set of messages.

- **SETUP**: This message enables a network client to request a connection from a source to a sink.
- **CALL_PROCEEDING**: This message is a response to the client at the source to acknowledge the setup request.
- **CONNECT**: This message is an acknowledgment of a successful trail creation.
- **RELEASE**: This message is used to tear down an existing trail.
- **RELEASE_COMPLETE**: This is used for acknowledging a release message.

The following sections and their corresponding tables define the messages in detail. Notice that every JIT signaling message contains the protocol version discriminator (PVD) field and the

---

[3] Notice that although our design borrows the ATM cell structure as the container for the JIT signaling messages, it does not require the actual existence of an ATM network to carry the messages. In fact any out-of-band hop-to-hop channel capable of transmiting 48 bytes atomicly will suffice.

TABLE I
SETUP MESSAGE

Message Information Field	Length (Octets)
Protocol Version Discriminator	2
Message Type	2
Origination Call Reference	4
Origination Address	8
Input Port Identifier	4
Input Wavelength Identifier	2
Bearer Signal Class	2
Client Layering Information	4
Destination Address	8
Cyclic Redundancy Check	2

TABLE II
CALL_PROCEEDING MESSAGE

Message Information Field	Length (Octets)
Protocol Version Discriminator	2
Message Type	2
Origination Call Reference	4
Output Wavelength Identifier	2
Estimated Initial Delay	4
Cyclic Redundancy Check	2

TABLE III
CONNECT MESSAGE

Message Information Field	Length (Octets)
Protocol Version Discriminator	2
Message Type	2
Origination Call Reference	4
Origination Address	8
Cyclic Redundancy Check	2

TABLE IV
RELEASE MESSAGE

Message Information Field	Length (Octets)
Protocol Version Discriminator	2
Message Type	2
Origination Call Reference	4
Origination Address	8
Cause	2
Cyclic Redundancy Check	2

Message Type field. The PVD field distinguishes the JIT signaling protocol from other signaling protocols that may be defined, and identifies the version of the protocol in use. The Message Type field in each message uniquely determines the exact layout of the message so that further decoding of the message can be easily accomplished. The message information fields are named and defined from the perspective of the receiver, i.e., the NE that processes the message. More detailed explanation of each of the message fields will be presented in Section VI-B. In the following message format tables, the length indicates the total length (in octets) of the message information fields. Since the JIT signaling messages are encoded within a 48-byte payload of a single ATM cell, they may or may not occupy the full 48 bytes. If it does not occupy the full cell, the remaining unused bits are set to zeros.

1) Setup: The SETUP message originates from a client requesting a connection to be set up across the network. Some of the information it carries are origination and destination addresses, origination call reference, input wavelength identifier, and bearer signal class, etc. (See Table I.) This message is passed from the originator to the next network element along the path toward the destination. On each hop a routing table is consulted to determine which port and wavelength it needs to use and creates a cross-connection between this connection point and the input connection point.

Observe that for the SETUP message sent from the source client to the edge switch, the input wavelength parameter specifies a preferred wavelength, and is *nonbinding* to the receiving switch. The receiving edge switch can subsequently demand the client to transmit in a different wavelength. In contrast, for the SETUP messages that are subsequently sent between switches, the input wavelength specified is binding to the receiving network element (switch or destination client). This is because we do not *require* wavelength translation capability to be present at each switch, and because of the JIT nature of the transmission. For intermediate switches which have the capability to change wavelengths, we expect wavelength selection to be a part of the routing decision.

2) Call_Proceeding: A CALL_PROCEEDING message is returned to the originator by the first network element as an acknowledgment of the setup request. It returns, among other things, an estimated initial delay time that the client should wait before transmitting its data. It also returns the actual output wavelength that the client should use to transmit its data burst. (See Table II.)

3) Connect: The CONNECT message is emitted by the destination. It is sent to the originator to signal a successful end-to-end connection after a call SETUP message is received at the destination. The call reference uniquely identifies the SETUP request that it acknowledges. When the originating client receives this message, it is assured that an connection path has been successfully set up end-to-end. (See Table III.)

4) Release: This message is sent either by the originating client, or the destination client, to tear down all cross connections along the path on an existing connection trail. It can also be sent by any network element to all network elements along the path to the originating client when it detects a SETUP failure. An intermediate switch along the connection trail uses the origination address and the origination call reference to identify the associated cross-connect to be torn down. (See Table IV.)

5) Release_Complete: This is a message acknowledging a cross-connect tear-down in response to the receipt of a RELEASE message. The message is sent to the RELEASE message sender after the cross-connect is torn down. (See Table V.)

B. Message Information Fields

Table VI summarizes the information elements that are utilized in the signaling messages. In the following description

TABLE V
RELEASE-COMPLETE MESSAGE

Message Information Field	Length (Octets)
Protocol Version Discriminator	2
Message Type	2
Origination Call Reference	4
Origination Address	8
Cyclic Redundancy Check	2

TABLE VI
MESSAGE INFORMATION FIELDS

Message Information Field	Reference
Protocol Version Discriminator	Tables 7 & 8
Message Type	Tables 9 & 10
Call Reference	Table 11
Address	Tables 12 & 13 & 14
Bearer Signal Class	Tables 15 & 16
Port Identifier	Table 17
Wavelength Identifier	Table 18
Delay	Table 19
Client Layering Information	Table 20
Cause	Table 21
Cyclic Redundancy Check	Table 22

of the message information field codings, the label "Reserved" means that the bits are reserved for possible future extensions. By convention, the caller/network shall set these bits to zero when sending the information element to the other side.

Unless indicated otherwise, a binary coded number has octets with lower octet numbers containing the more significant bits. The octets with the lowest number contain the most significant bits, and the octet with the highest octet number contains the least significant bits. Within an octet, bits with the higher bit number contain the more significant bits.

1) Protocol Version Discriminator: The protocol discriminator for MONET JIT signaling is shown in Table VII. The PVD field distinguishes the JIT signaling protocol from other signaling protocols that may be defined, and identifies the version of the protocol in use. The detailed coding of protocol and version discriminator field for the MONET JIT-SP prototype implementation is given in Table VIII.

2) Message Type: Table IX shows the message type information field. The Message Type field in each message uniquely determines the exact layout of the message so that further decoding of the message can be easily accomplished. The detailed coding of message type for JIT signaling is given in Table X.

3) Call Reference: A JIT SETUP message includes an origination call reference field as shown in Table XI. When originating a new call, the caller shall allocate a new call reference value associated with the call. A call reference value remains fixed for the duration of the call. It is a four-byte unsigned integer that is assigned sequentially with wrap around, thereby allowing sufficient time for timeouts and retries to cover for lost messages. When combined with the originator's address, the origination call reference can uniquely identify a JIT connection call.

TABLE VII
PROTOCOL DISCRIMINATOR

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Protocol Discriminator							
2	Version Discriminator							

TABLE VIII
PROTOCOL VERSION DISCRIMINATOR

Octets	Bits							
	8	7	6	5	4	3	2	1
1	0	0	0	1	0	0	0	1
2	0	0	0	0	0	0	0	1

TABLE IX
MESSAGE TYPE

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Message Type Code							
2	Message Type Code cont'd.							

TABLE X
MESSAGE TYPE ENCODING

Octets 1 & 2	Meaning
0111000000000000	SETUP
0000111000000000	CALL-PROCEEDING
0000000111000000	CONNECT
0000000000111000	RELEASE
0000000000000111	RELEASE-COMPLETE

4) Address: The MONET JIT signaling addressing scheme is designed to be extensible. Table XII shows the address information field layout for a unicast address. In addition to the unicast address class encoding, the remaining six octets of the field consist of a triplet of network, host, and port identifiers. Table XIII shows the address class encoding for a unicast address that has been defined and is being used in the MONET implementation. The Network, NE, and Port identifiers are all 16-bit unsigned numbers. Likewise, Table XIV shows the address class encoding for multicast address. However the detailed multicast address format has not been defined and is reserved for future specification.

5) Bearer Signal Class: The bearer signal class is described in the Tables XV and XVI. The Type bit (Octet 1, Bit 8) indicates whether the payload bearer signal is an analog (0) or a digital (1) signal. (See Tables XV and XVI.) The subsequent three-byte parameter is divided according to its type. For an analog signal, octet 2 indicates the signal-to-noise ratio (SNR). Octet 3 and 4 indicate its dynamic range. For a digital signal, octet 2 to 4 holds a three-byte clock (i.e., data) rate expressed with a 10 kb/s resolution.

6) Port Identifier: The JIT SETUP message includes an input port identifier field encoded as a two-byte unsigned

TABLE XI
CALL REFERENCE

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Call Reference value							
2	Call Reference cont'd.							
3	Call Reference cont'd.							
4	Call Reference cont'd.							

TABLE XII
ADDRESS ENCODING

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Address Class							
2	Reserved							
3	Network ID							
4								
5	Network Element ID							
6								
7	Port ID							
8								

TABLE XIII
ADDRESS CLASS ENCODING FOR UNICAST

Octets	Bits							
	8	7	6	5	4	3	2	1
1	0	0	0	0	0	0	0	0

TABLE XIV
ADDRESS CLASS ENCODING FOR MULTICAST

Octets	Bits							
	8	7	6	5	4	3	2	1
1	1	1	1	1	1	1	1	1

TABLE XV
ANALOG SIGNAL TYPE ENCODING

Octets	Bits							
	8	7	6	5	4	3	2	1
1	0	Reserved						
2	SNR							
3	Range							
4	Range cont'd							

TABLE XVI
DIGITAL SIGNAL TYPE ENCODING

Octets	Bits							
	8	7	6	5	4	3	2	1
1	0	Reserved						
2	Clock Rate							
3	Clock Rate cont'd							
4	Clock Rate cont'd							

TABLE XVII
PORT IDENTIFIER

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Port Identifier							
2	Port Identifier cont'd							

TABLE XVIII
WAVELENGTH IDENTIFIER

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Wavelength Identifier							
2	Wavelength Identifier cont'd							

TABLE XIX
DELAY

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Delay							
2	Delay cont'd							
2	Delay cont'd							
2	Delay cont'd							

integer as shown in Table XVII. It is used to uniquely identify the input port at a WDM switch that will receive the transmitted data burst. This port identifier is a number assignment that is local to the receiving WDM switch.

7) Wavelength Identifier: The JIT SETUP and CALL_PROCEEDING messages both include a wavelength identifier field defined as follows: Two bytes of unsigned integer (N). It has a 50 GHz channel separation so that lambda $= N \times 50$. (See Table XVIII.) The field size and frequency resolution of the bits were determined to permit use on wireless systems as well, for optical networking-wireless network interoperability.

8) Delay: The estimated initial time delay field included in the SETUP message is a four-byte unsigned integer value with a microsecond unit. (See Table XIX.)

9) Client Layering Information: The client layering information field in the SETUP message encodes the higher layer protocols in the protocol stack of the transmitted data burst. The rationale for including such information in the setup message is to allow the receiver either to setup the necessary protocol stack to process the incoming message, or to verify upfront if it does not have the protocol handling capability to process the incoming message. The client layering information field has a total of 14 bytes, allowing for a maximum of seven client layers, with two bytes of information per layer. The first bit indicates whether it is standard or user defined protocol. We encourage a broad definition of "protocol" to include compression and encrytion services at or between the layers. The detailed protocol encoding has not been defined, and is reserved for future specification. (See Table XX.)

TABLE XX
CLIENT LAYERING INFORMATION

Octets	Bits							
	8	7	6	5	4	3	2	1
1	STD/User Defined	Protocol Encoding						
2	Protocol Encoding cont'd							
3	STD/User Defined	Protocol Encoding						
4	Protocol Encoding cont'd							
5	STD/User Defined	Protocol Encoding						
6	Protocol Encoding cont'd							
7	STD/User Defined	Protocol Encoding						
8	Protocol Encoding cont'd							
9	STD/User Defined	Protocol Encoding						
10	Protocol Encoding cont'd							
11	STD/User Defined	Protocol Encoding						
12	Protocol Encoding cont'd							

10) Cause: The JIT RELEASE message includes a "cause" information field that encodes the reason or condition that triggers the release request. It is encoded in two bytes as depicted in the following Table XXI. The remaining bits are reserved for possible future extension. Below we summarize the meaning of each of the defined bits.

- *Normal Termination:* "1"means that the release request is the result of a normal connection termination; "0" means otherwise.
- *Source Abort:* "1" means the release is triggered by source abort action e.g., upon detecting a timeout as a result of a message lost; "0" means otherwise.
- *Persistent Condition:* It indicates whether the condition that triggers the release message is likely to be a persistent (1) condition, or is a transient (0) condition.
- *Destination Unknown:* "1" means that the destination address is unknown; "0" means otherwise.
- *Output Channel Unavailable:* "1" means that an intermediate switch cannot secure an output channel to propagate the setup request and its data burst.
- *Clock Rate Too High:* "1" means that the digital bearer signal has a data rate that is too high for the switch, the conditioning of the outgoing fiber, or the end recipient; "0" means otherwise.
- *Digital Only:* "1" means that the intermediate switch can only switch a digital signal while the bearer is analog; "0" means otherwise.
- *Protocol Incompatible:* "1" means the setup request is not compatible with the signaling implementation at an intermediate switch; "0" means otherwise.
- *Destination Reject:* "1" means the request was rejected by the destination, e.g., because it does not have the ability to process the indicated client layer information; "0" means otherwise.
- *Invalid Global Call Reference:* "1" means that the request has an invalid global call reference value (i.e., combination of origination address and call reference); "0" means otherwise.

11) Checksum: We have selected a 16-bit CRC scheme to test for data corruption of JIT signaling messages (see Table XXII). In cyclic redundancy checking, the bit strings (16-bit strings) are treated as representations of polynomials with coefficients of zero and one. A 16-bit message is regarded as coefficient list for a polynomial with k terms, ranging from X_{k-1} to X_0. When the polynomial code method is employed, the sender and receiver must agree upon a generator polynomial. We can use either CRC-CCITT or CRC-16; both are 16-bit polynomials. The basic idea is to append a checksum to the end of the message in such a way that the polynomial represented by the checksummed message is divisible by the generator polynomial. When the receiver gets the checksummed message, it tries to divide it by the same generator polynomial. If there is a remainder there has been a transmission error.

A 16-bit checksum such as CRC-16 or CRC-CCITT, catches all single and double bit errors, all errors with an odd number of bits, and burst errors of length 16 or less, 99.997% of 17-bit burst error, and 99.998% of 18-bit and longer bursts [25]. In MONET, we have implemented CRC-16.

VII. MONET JIT SIGNALING IMPLEMENTATION

MONET provides out-of-band ATM channels which are used to carry the JIT signaling messages. These are the same ATM channels used as the data communications channel (DCC) in the MONET network control and management (NC&M) system [23]. The ATM-based DCCs are carried as an embedded wavelength over each MONET link and is independent of the payload wavelengths used to carry user data. In MONET the embedded DCC wavelength used is the ITU standard signaling wavelength at 1510 nm, which is below the payload wavelengths. The DCC wavelengths are being dropped at each network elements. As described in Section VI the MONET signaling messages are carried in AAL0 over the embedded ATM channels (see Fig. 9).

In MONET, we developed a prototype implementation of the JIT-SP software and tested it against the Lucent and Tellium WDM network elements (NEs) deployed in the Washington, DC testbed network. The JIT signaling software is integrated as a component of the WDM agent implementation [8], as is illustrated in Fig. 10. The main components of the agent are the CORBA interface to NC&M, JIT signaling, and NC&M-to-NE interface session manager. The CORBA interface is responsible for representing the logical components of the NEs such as its equipment hierarchy and facility objects such as fabric, cross-connect objects and termination points. These objects are exposed to the NC&M for equipment inventory and cross-connect provisioning. The south-bound interface is a session manager connection to the native NE controller through a TCP socket. This interface has directives to get equipment, create cross-connect and delete cross-connect, etc. It is also responsible for servicing multiple threads of requests from provisioning and signaling. Last, the JIT signaling component handles signaling requests coming from neighboring NEs and JIT clients/hosts. The JIT signaling component is composed of several modules: JIT engine, signaling path, encoder, decoder, and router.

The core of the signaling component is the JIT signaling engine. This module accepts decoded signaling messages from its immediate neighbors. It decides upon which actions to take depending on the type of message described earlier. For example,

TABLE XXI
CAUSE

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Normal Termination	Source Abort	Persistent Condition	Destination Unknown	Output Channel Unavailable	Clock Rate Too High	Digital Only	Protocol Incompatible
2	Destination Reject	Invalid Global Call Reference	Reserved					

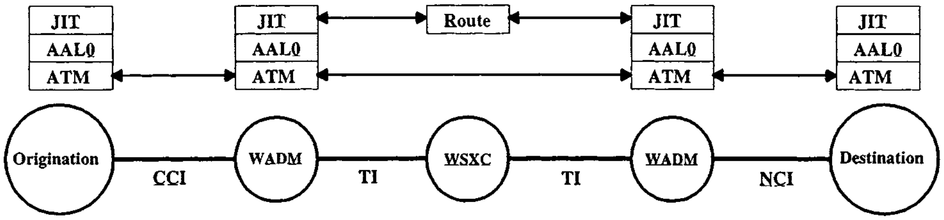


Fig. 9. JIT signaling architecture for multiwavelength optical networks.

TABLE XXII
CYCLIC REDUNDANCY CHECK

Octets	Bits							
	8	7	6	5	4	3	2	1
1	Cyclic Redundancy Check							
2	Cyclic Redundancy Check cont'd							

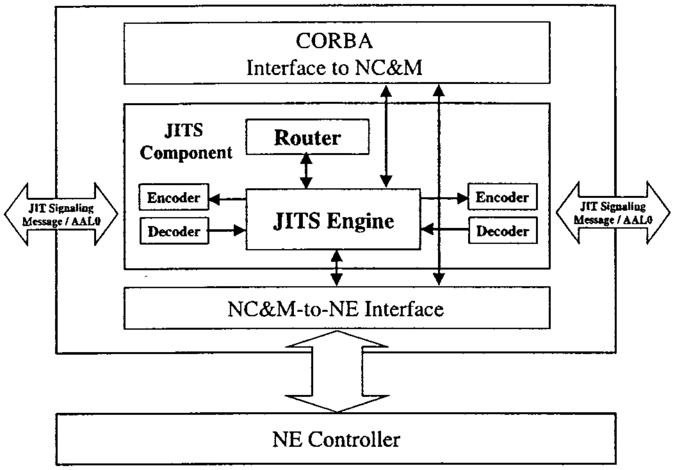


Fig. 10. JIT signaling internal functional architecture.

if it receives a SETUP request, it will look at the destination address and consult the router module for the best next hop (which can be as simple as a content adressable memory hardware implementation), validates the needed resources, instructs the NE to make the cross-connect, and finally sends SETUP request to the next hop neighbor through a signaling path. The JIT engine implements the signaling algorithm, and will handle all the basic scenarios described earlier in Section V, such as connection setup, connection tear-down, failed connection due to message loss, and failed connection due to rejection.

The signaling path implements the communication channel between NE neighbors. For each neighboring NE and client/host, the signaling module shall create an instance of signaling path to establish communication with them. Signaling paths have two types, user network interface (UNI) and network node interface (NNI), that define the connection boundaries to the engine. Each signaling path is composed of a decoder and an encoder. These components translate signaling structures into message bytes as described in Section VI. The message format are placed into a single ATM cell payload and sent as an AAL0 ATM cell. The decoder performs the reverse by looping at the signaling path input, performing a CRC, and converting the AAL0 payload into a message structure. These messages are then passed onto the JIT engine.

The JIT router implements a simple next hop resolution routing algorithm. Because routing is not the focus of our experiments we did not create a more elaborate routing schemes with flooding and route propagation. Instead, we opted to implement simple static routes. Because the MONET testbed is small, static routes are manageable and sufficient for our experimentation purposes. More elaborate routing schemes are necessary to deploy JIT signaling over a large network. Routing for JIT switching and signaling is an open issue reserved for subsequent study.

VIII. CONCLUSION

JIT optical burst switching is a novel switching paradigm that combines the desirable features of circuit and packet switched schemes for optical WDM networks.

Under the MONET project, a JIT signaling protocol has been designed and implemented. It features out-of-band control signal processing that eliminates buffering of data-burst at intermediate nodes, while minimizing the setup time, and maximizing the cross-connect bandwidth efficiency. The signaling protocol software implementation itself was tested and validated in the MONET Washington, DC testbed network [3]. Additional testing and experimentation of JIT optical burst signal transmission are scheduled in a follow-on program with participation from Telcordia and several government agencies including the Laboratory for Telecommunications Science and the Naval Research Laboratory. These experiments will provide the opportunity to gain further experience and insight with the JIT-OBS approach. The objective is to identify the technical hurdles that must be overcome, and to propose solutions so as to develop OBS into a practical and efficient optical switching technology.

As it was noted in [26], "…in the future, bandwidth will not be our problem. Latency will be the major challenge to overcome." Toward meeting this challenge, JIT optical burst switching provides a good mechanism for ultra-low latency transport of variable sized bursts of data across an optical WDM network.

ACKNOWLEDGMENT

The authors wish to thank Y. Tsai, Telcordia Technologies, and R. Ramamurthy, then at Telcorida and currently at Tellium Inc., for their help in the analysis of the JIT optical burst switching performance. We wish to thank J. Parker, then at Optical Networks, Inc., for helpful discussions during the definition stage of the JIT signaling message design. We also wish to thank J. Pastor, Telcordia Technologies, for his help in prototyping and testing the JIT signaling protocol implementation.

"The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official polices, either express or implied, of the Army Research Laboratory or the U.S. Government."

REFERENCES

[1] R. E. Wagner, R. C. Alferness, A. A. M. Saleh, and M. S. Goodman, "MONET: Multiwavelength optical networking," *J. Lightwave Technol.*, vol. 14, no. 6, pp. 1349–1355, June 1996.

[2] "The MONET New Jersey network demonstration," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1199–1219, Sept. 1998. The MONET Research Team.

[3] A. V. Lehmen, "Large-scale demonstration of multiwavelength optical networking in MONET Washington, DC network," in *OFC'99*, vol. 3, San Diego, California, Feb. 1999, p. 237.

[4] W. T. Anderson *et al.*, "The MONET project—A final report," in *OFC'2000*, vol. 2, Baltimore, MD, Mar. 2000, pp. 148–149.

[5] J. Y. Wei, C.-C. Shen, B. J. Wilson, and M. J. Post, "Network control and management of a reconfigurable WDM network," in *MILCOM'96*, vol. 2, McLean, VA, Oct. 1996, pp. 581–586.

[6] Y. Tsai, J. Wei, C.-C. Shen, B. Wilson, M. Post, M. Maeda, J. Pastor, and A. Luss, "A distributed network management system for reconfiguration WDM networks," presented at the NFOEC'97, Sept. 1997.

[7] J. Wei, M. Post, C.-C. Shen, B. Wilson, J. Pastor, M. Maeda, and Y. Tsai, "Network control and management of reconfigurable WDM all-optical network," in *NOMS'98*, vol. 3, New Orleans, LA, Feb. 1998, pp. 880–889.

[8] J. Y. Wei, C.-C. Shen, B. J. Wilson, M. J. Post, and Y. Tsai, "Connection management for multiwavelength optical networking," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1097–1108, Sept. 1998.

[9] L. Kleinrock, "New technologies: Bringing you up to gigabit speed," presented at the Gigabit Networks: Emerging Commercial Applications and Opportunities Conference, Washington, DC, July 1992.

[10] [Online]. Available: http://www.eecis.udel.edu/ mills/highball.htm.

[11] (1995, Nov.) Contract with the DoD, Palo Alto, CA [Online]

[12] G. Hudek *et al.*, "Signaling analysis for an all-optical network that uses burst switching," presented at the OFC'95, San Diego, CA, Feb. 1995.

[13] J. Y. Wei and Y. Tsai, "Signaling protocols for optical WDM switching," in *Proc. SPIE Conf. All-Optical Networking: Architecture, Control and Management Issues*, vol. 3531, Boston, MA, Nov. 1998, p. 465.

[14] M. Yoo, M. Jeong, and C. Qiao, "A high-speed protocol for bursty traffic in optical networks," in *SPIE Conf. All-Optical Networking: Architecture, Control and Management Issues*, vol. 3230, Boston, MA, Nov. 1997, pp. 79–90.

[15] C. Qiao and M. Yoo, "Choices, features and issues in optical burst switching," *Optical Networks*, vol. 1, no. 2, pp. 36–44, Apr. 2000.

[16] J. S. Turner. Terabit burst switching. [Online]. Available: http://www.arl.wustl.edu/jst/projects/TerabitBurst.html

[17] G. K. Chang *et al.*, "A proof of concept, ultra-low latency optical label switching testbed demonstration for next generation Internet networks," in *OFC'2000*, vol. 2, Baltimore, MD, Mar. 2000, pp. 56–58.

[18] D. Awduche *et al.*, Multiprotocol lambda switching: Combining MPLS traffic engineering control with optical cross-connects, in IETF Internet Draft, draft-awduche-mpls-te-optical-01.txt, Nov. 1999.

[19] D. Saha *et al.*, RSVP extensions for signaling optical paths, in IETF Internet Draft, draft-saha-rsvp-optical-signaling-00.txt, Mar. 2000.

[20] Y. Fan *et al.*, Extensions to CR-LDP and RSVP-TE for optical path setup, in IETF Internet Draft, draft-fan-mpls-lambda-signaling-00.txt, Mar. 2000.

[21] F. Masetti *et al.*, "High-speed high-capacity ATM optical switches for future telecommunication transport networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 979–998, June 1996.

[22] D. Chiaroni *et al.*, "All-optical fast packet-switched networks: Physical and logical limits of operation," in *SPIE Conf. All-Opt. Networking: Architecture, Contr., Management Issues*, vol. 3531, Boston, MA, Nov. 1998, pp. 378–389.

[23] J. Y. Wei, "The role of DCN in optical WDM networks," in *OFC'2000*, vol. 4, Baltimore, MD, Mar. 2000, pp. 142–144.

[24] J. Y. Wei, J. L. Pastor, R. S. Ramamurthy, and Y. Tsai, "Just-in-time optical burst switching for multiwavelength networks," in *IFIP Broadband Commun.*, Hong Kong, Nov. 1999, pp. 339–352.

[25] R. B. Wells, *Applied Coding and Information Theory for Engineers*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[26] D. Farber, "Predicting the unpredictable—The impact of all-optical networking," presented at the Keynote Speech at the IEEE LEOS Summer Topical Workshop on Broadband Networking, Monterey, CA, July 1998.

John Y. Wei (M'00) received the Ph.D. degree in computer science from the California Institute of Technology, Pasadena.

He is the Director of the Broadband Network Management Research Group at Telcordia Technologies, Red Bank, NJ. He is leading a research group in developing the next-generation network operations and management systems architecture and prototype for broad-band networks. His research group is responsible for the network management research tasks in the DARPA funded projects of ATDNet, MONET, and NGI, which focus on the network control and management issues of broad-band networks covering technologies from ATM, SONET, optical WDM, to IP over WDM. He was the task leader for the NC&M work in MONET, and the NGI Optical Label Switching Project, as well as the current Principal Investigator of the NGI SuperNet NC&M Project. He joined Bellcore (now Telcordia Technologies) in 1989 and has worked on network management, distributed systems, and fault-tolerant computing.

Dr. Wei is a member of the ACM.

Ray I. McFarland, Jr., (M'80) received the B.S. degree in engineering (with high honors) with specialization in computer engineering from Case Institute of Technology, Cleveland, IN, in 1971 and the M.S. degree in computer science from the University of Maryland, College Park, in 1976.

Since 1971, he has been with the Department of Defense, Fort Meade, MD. He is currently assigned to the Laboratory for Telecommunication Science, where he is a Member of Senior Technical Staff in Advanced Basic Network Technology. His research interests include high-performance networks, specifically with Tb/s aggregate link bandwidth and up to 100 Gb/s individual connection data rates, switched optical networks, network and internetwork architectures and protocols, network and internetwork management, and ubiquitous computing and distributed systems.

Mr. McFarland is a member of the IEEE Computer and Communications Societies, Tau Beta Pi, Theta Tau, the Association for Computing Machinery and its Special Interest Groups for Communications and for Operating Systems, and the Armed Forces Communications and Electronics Association. He was the Cochair of the IEEE LEOS "Summer Topical on Broadband Optical Networks and Technologies, an Emerging Reality" and Chair of the Networking Program Subcommittee of the 1997 OSA/IEEE Optical Fiber Communications Conference.