An Intra- and Inter-Domain Routing Architecture for Optical Burst Switched (OBS) Networks

Ilia Baldine[1], Pronita Mehrotra, George Rouskas[2], Arnold Bragg[1], and Dan Stevenson[1]

[1] Center for Advanced Network Research
RTI International, Inc.
3040 Cornwallis Road, PO Box 12194
Research Triangle Park, NC 27709-2194

[2] Computer Science Department
North Carolina State University
Box 7534
Raleigh, NC 27695-7534

Abstract – **We describe an intra- and inter-domain routing architecture for Just-In-Time (JIT) optical burst switched (OBS) networks. The architecture addresses the problem of routing optical signals of varying types across an all-optical burst-switched backbone network while maintaining the optical signal quality required by each application. The architecture distinguishes between routing for bursts and ancillary signaling messages ("data plane routing"), and routing for other management and control messages ("control plane routing").**

I. Introduction

A fundamental element of the *JumpStart* optical burst switching (OBS) architecture and its Just-in-Time (JIT) protocol suite is the functional separation between the *data* and *control* planes. The all-optical (OOO) data plane is responsible for transporting data bursts between endpoints. The opto-electronic (OEO) control plane is an unreliable packet-switched overlay, and is responsible for all signaling – including path establishment and release for data bursts, routing (e.g., path computation, reachability, availability), and other network management and control functions.

Figure 1 illustrates the dichotomy. Each OBS node contains an optical cross-connect (OXC), and a JIT Protocol Acceleration Circuit (JITPAC) that implements the JIT protocol suite in hardware and controls the OXC. The data plane consists of the OXCs and data channels on the fiber links that interconnect OXCs. The control plane consists of the JITPACs and the dedicated, out-of-band signaling channel that interconnects them. The topologies of the data and control planes are identical; the control plane's signaling channel occupies one wavelength on the same fiber as the data channels. (The *JumpStart* architecture and JIT signaling protocol, message format, and addressing scheme are described in [1-5].)

There are two types of JIT signaling messages: **(1)** messages associated with the transmission of data bursts, and **(2)** all other management and control messages. Data bursts and their ancillary signaling messages must follow the same end-to-end path. E.g., a JIT **`SETUP`** message precedes each data burst and is responsible for establishing the burst's path, so it must touch each JITPAC (and OXC) in the path. However, there is no requirement that other management and control messages – i.e., signaling not associated with burst transmission such as messages that exchange routing information or report network outages – must follow the same paths as bursts. As a result, we have developed two independent routing implementations – one for transmission of bursts and ancillary signaling messages (*data plane routing*), and a second for all other JIT management and control messages (*control plane routing*).

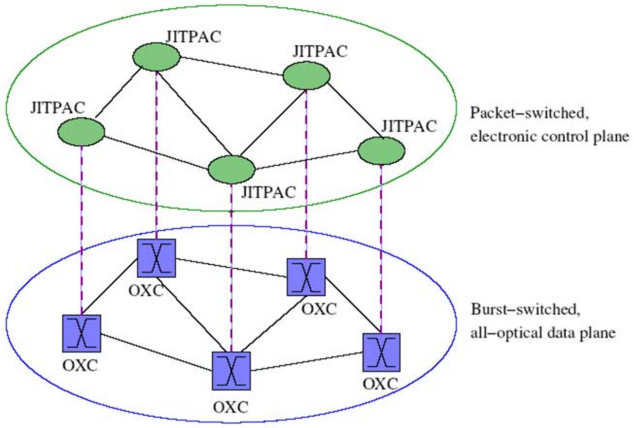

Figure 1: *JumpStart* architecture with data and control planes.

In Section II, we explain the rationale for the two routing implementations. We describe how these implementations are used to support intra-domain routing in Sections III and IV, and inter-domain routing in Sections V and VI. We describe two new JIT resource provisioning variants in Section VI, and conclude in Section VIII.

II. Two Independent Routing Implementations

The decision to support two independent routing implementations was motivated by the following observation. A transparent burst-bearing OOO path between OBS

0-7803-9277-9/05/$20.00/©2005 IEEE

endpoints must satisfy a completely different set of service quality requirements than an OEO path that carries JIT control messages between the same endpoints. The information required for routing bursts is very different from the information required for routing control messages. Having two routing implementations allows each to be optimized for its specific requirements.

A. *Scope – Intra- and Inter-Domain Routing*

A *JumpStart* OBS network consists of some number of interconnected optically-transparent *domains*. (Figure 1 comprises a single domain.) Each domain is responsible for routing traffic within it and across it. A domain contains of some number of OXCs (controlled by JITPACs) that are interconnected with other OXCs and client nodes, creating a topological mesh. Each OXC has some number of multi-wavelength input ports, output ports, a wavelength-selective crossbar, and service circuits that are capable of amplifying a signal, compensating for distortion, converting from one wavelength to another, etc. (G, DC and W in Figure 2). The *JumpStart* architecture does not require fiber delay lines.

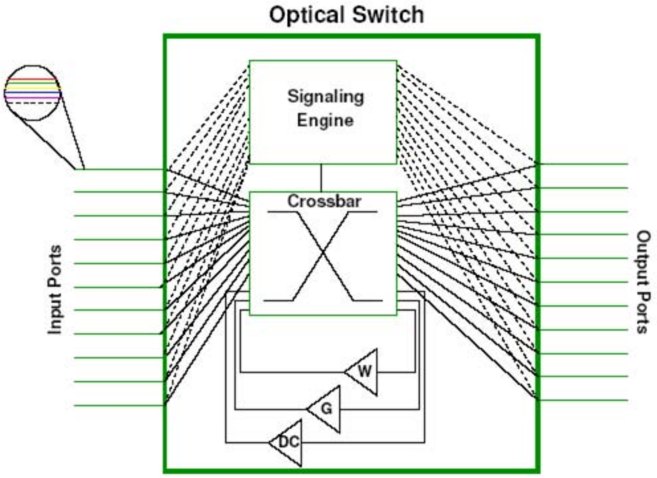

Figure 2: Reference switch architecture with service circuits.

Data plane routing and control plane routing are either *intra-domain*, with signaling and routing data confined to a single domain; or *inter-domain*, with exchanges of data and data-transmission signaling between domains. Intra-domain routing assumes that the domain is optically transparent; i.e., that either its equipment is data-format insensitive (via OOO amplifiers or optical 2R), or that it provides OEO conversion points between its islands of transparency to support all data formats.

It is unlikely that the interconnects between domains are optically transparent, so JIT signaling is terminated and reinitiated at domain boundaries for inter-domain routing. Various types of information can be exchanged between domains, depending on the level of trust. E.g., domains may only exchange information to support JIT signaling and routing, or they may share complete information about their topologies and the characteristics of their links and switchgear.

B. *Components – Forwarding and Path Computation*

Data plane routing and control plane routing each have two distinct components. The *forwarding* component is responsible for transporting data and control messages from source to destination across a single OBS node. Each node maintains two independent forwarding tables – a *burst forwarding table* for data bursts and ancillary signaling messages (e.g., **`SETUP`**), and a *control forwarding table* for other control messages. The forwarding component uses the appropriate forwarding table and information from control messages to make forwarding decisions for each plane.

The *path computation* component is responsible for the construction and maintenance of forwarding tables for each implementation. It consists of one or more routing protocols that collect and exchange routing-related information among nodes, and algorithms that convert this information into forwarding tables. Separate protocols are used for burst and control topology discovery, link status updates, and path computation for the two implementations.

C. *Control Plane Routing Objective*

The control plane's routing objective is to compute shortest paths between OBS nodes to support the efficient exchange of management and control messages. This is similar to routing objectives in conventional OEO, unreliable, packet-switched networks that use a link state protocol like OSFP to exchange routing information (e.g., link and node states, reachability). The control plane is OEO, and is not concerned with *optical service quality* (OSQ).

D. *Data Plane Routing Objective*

The data plane uses the transparent OOO OBS backbone, so its routing objective is to compute paths that guarantee the OSQ of bursts between OBS endpoints. This is similar to routing objectives in constraint-based routing and wavelength assignment (RWA) problems, which are more complex than shortest-path algorithms and which require more information than link state and reachability.

A *JumpStart* OBS network satisfies client requests for a particular OSQ for data bursts by choosing appropriate links through which the signal is routed, and by engaging service circuits as required (Figure 2). E.g., if the shortest available path cannot satisfy the requested OSQ and if no service circuits are available on that path, then the network may select a longer alternate route with service circuits. This allows the network to condition the burst's signal so that it arrives at the destination with the requisite OSQ.

OSQ-aware forwarding and routing are challenging for several reasons: optical networks are not homogeneous with respect to signal quality; optical fiber does not behave as an ideal channel, especially at bit rates above 10 Gbit/s where a number of effects become more noticeable [11-15]; network elements also contribute to signal distortion; and operational OBS networks will likely have relatively large optically transparent domains, a disparate mix of fiber and switchgear, and will transport digital and analog signals at various rates between 2.5 Gbit/s and 160 Gbit/s. This will require sophisticated OSQ-aware mechanisms for forwarding and routing within domains that lack per-hop OEO conversions.

E. *Implementation Issues*

Having two routing implementations enforces the functional separation between the data and control planes, so that modifications or extensions to one will have little or no impact on the other. Decoupling the data and control planes also reduces the complexity of the routing architecture, and makes it easier to deploy by (re-)using proven protocols and algorithms when appropriate.

III. Intra-Domain Control Plane Routing

We use an OSPF-like link state protocol to compute and establish shortest paths between nodes to support the efficient exchange of JIT management and control messages [6]. Extending an existing, proven protocol (rather than developing a new one) makes it easier to guarantee correct, consistent, and robust operation under a wide range of deadlock, livelock, and failure scenarios. We have adapted JITPAC controllers to support routing (denoted "JITPAC-R" in Figure 3).

Each node broadcasts link state advertisements (LSAs) to all other nodes in its domain over the out-of-band OEO signaling channel. LSAs contain information about the status and attributes of the control interfaces of the node where they originate. LSAs *only* contain information relevant to routing in the control plane; routing information for the data plane is different, and is disseminated in a different way (Section IV).

LSAs are broadcast using reliable flooding (Figure 3), which guarantees that every node in the domain receives a copy of each LSA even in the presence of node or link failures as long as the domain remains connected [6]. Each node obtains complete information about the topology of the control plane, stores the information in its control plane routing database, and runs a variant of Dijkstra's shortest-path-first (SPF) algorithm locally (and independently of other nodes) to determine how to reach other nodes in its domain. Link costs are based on an administratively-defined metric. Each node updates its control forwarding table to reflect the computed shortest paths.

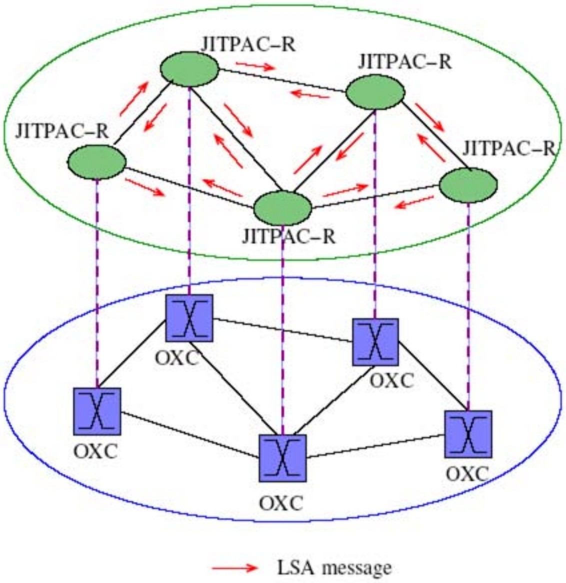

Figure 3: *JumpStart* intra-domain control plane routing.

Each node broadcasts an LSA whenever there is a change in the status of its control interfaces. This ensures that each node not only establishes shortest paths to every other node in its domain over the control overlay, but also that the paths are periodically updated to reflect current state. If paths fail or become suboptimal due to a change in status in a control interface or link, they will eventually be updated via the LSA broadcast mechanism and the path computation that it triggers. Using a multicast variant of the link state protocol (e.g., based on MOSPF) enables each node to compute multicast control path trees in addition to point-to-point paths.

IV. Intra-Domain Data Plane Routing

A. *Approach*

We have developed a semi-centralized (or weakly distributed) routing architecture for computing paths within a single *JumpStart* domain. The architecture:

(1) Supports quality of service (QoS) requests from operators, users, and applications – e.g., bandwidth or rate, latency, jitter, error rate, dynamic range (for analog signals), etc. These may be represented as a small number of application QoS classes.

(2) Supports a small number of OSQ classes (e.g., bronze, silver, gold, platinum).

(3) Maps QoS requests (or classes) from operators, users, applications onto the optical layer's OSQ classes.

(4) Estimates end-to-end OSQ via empirical measurements and/or derivations of optical plane static and dynamic effects. OSQ parameters are carried by control messages on the signaling channel, and are updated at each node along the path. The parameters provide a quantitative estimate of OSQ at each node and at the destination. OSQ parameters are applicable to all kinds of traffic, as the transparent data plane makes no assumptions about whether traffic is analog or digital, whether it is constrained to certain modulations or data rates, etc.

(5) Maps the effects onto the OSQ classes, and uses each OSQ class' routing algorithm for burst path calculations and burst forwarding algorithms.

B. OSQ Metrics

Optical signal-to-noise ratio (OSNR) and optical jitter (O-jitter) are two important OSQ metrics. OSNR measures the ratio of signal power to noise power at the destination. End-to-end OSNR is a function of many physical layer impairments, all of which degrade the quality of the signal as it propagates through the all-optical network. A majority of these impairments can be measured or derived on a link-by-link basis. Link impairment estimates are coupled with estimates of impairments induced by optical devices (OXCs, EDFA amplifiers), and these estimates are used by the routing and forwarding algorithms.

O-jitter is the distortion of the optical signal representing a symbol in the temporal domain, which may introduce errors at the destination. O-jitter is also a function of linear and non-linear impairments induced by various dispersion and phase-modulation effects.

Note that application QoS – e.g., bandwidth, latency, jitter, error rate, dynamic range – cannot be determined solely by OSQ. The bandwidth seen by an application also depends on the instantaneous transmission rate, the burst inter-arrival rate, and the burst blocking rate. Application latency and jitter also depend on the transport protocol used by the application, queuing effects introduced by burst assembly and scheduling, burst route pinning, etc.

Dynamic range and end-to-end bit error rate (BER) cannot be determined solely from the optical layer, as both are evaluated within the electrical plane [7-9]. BER is used as a QoS parameter in IP networks, but it is not an appropriate parameter for JIT OBS networks because it is specific to digital transmissions, and because it is not feasible to have BER analyzers at destination nodes to determine whether the requested QoS was achieved. However, it is possible to *estimate* BER from optical measures.

C. Using OSNR and O-Jitter to Estimate BER

The first way to estimate OSQ is to view end-to-end OSNR and O-jitter as the optical equivalents of electrical SNR and jitter, and to use them to compute total BER [7,9]. As noted, OSNR and O-jitter are functions of various optical layer linear and non-linear impairments that can be measured or derived for each link and each optical device along the path, and used by the routing and forwarding algorithms.

An application specifies a BER, which is converted to an OSNR and O-jitter budget at the source node and embedded in the burst's **SETUP** message. As the **SETUP** traverses the burst's path, each node adjusts the embedded OSNR and O-jitter budgets based on its link quality and what the link may be carrying on other channels. E.g., a **SETUP** message with an initial OSNR budget of 20 db might traverse a link that degrades the OSNR by 3 db. The node that routes the **SETUP** message updates the OSNR budget to 17 db before sending it out over the output link. If the OSNR or O-jitter budgets reach zero before arriving at the destination node, the **SETUP** message is dropped as the requested OSQ cannot be met.

OSNR and O-jitter do not give an indication of whether gain compensation or dispersion compensation is required, so additional parameters are needed for different types of compensation devices (Table 1).

Determining OSNR and O-jitter budgets from the requested BER in not straightforward. In general, BER can be split into OSNR and O-jitter budgets in an infinite number of ways. It is not clear what criteria should be used in determining the optimal split.

Table 1: OSQ parameters required to estimate end-to-end BER; first approach.

OSQ Parameter	
OSNR	Optical signal-to-noise ratio
O-jitter	Optical jitter
Power	Total power in the channel
CD	Chromatic dispersion
PMD	Polarization mode dispersion
--	Other compensation-based parameters

Another problem is that forwarding becomes computationally intensive because it is not possible to determine OSNR solely by measurement. OSNR measurements determine the power in each channel, which can be corrupted due to non-linear effects. So each node must compute various distortion components at each link and combine them in some way to get the total distortion, which is then used to update the budgets. The computation depends on the traffic dynamics on each channel at each node.

D. Using Static/Dynamic Parameters to Estimate BER

The second way to estimate OSQ is to update and forward static and dynamic parameters rather than OSNR and O-jitter budgets. Static parameters correspond to the linear and non-linear parameters of fiber, and dynamic parameters correspond to the effects of traffic carried on different channels. Since the dominant effects are caused by wavelength (ω) and intensity ($|E|$) dependencies of the refractive index (*n*), we can use parameters that define these effects to any desired order by splitting the refractive index into two separate functions and use these functions to define a number of static parameters. We can expand this model to include polarization mode dependence and thus account for polarization mode dispersion and polarization dependent losses. Some parameters are provided by fiber manufacturers.

With this approach, as the **SETUP** message traverses the network, only the static and dynamic parameters need to be adjusted. E.g., a burst transiting two links (of lengths l and l') with a particular static parameter having values x and x' is indistinguishable (from a distortion perspective) from a burst traversing a single link of length ($l + l'$) with equivalent parameter y. So when a **SETUP** message arrives with a first-order wavelength dependence $\partial n/\partial w$ value of A, then the length of the outgoing link and the distance that the message has already traversed can be used to compute an effective value of A' for the entire distance.

The forwarding component must also decide whether to route the burst through a service circuit. If the power level falls below a certain threshold then burst must pass through an amplifier. This can be determined by monitoring power levels, or by computation. A simple metric like $D \times L$ (where L is distance traversed) can be used to determine whether a burst requires dispersion compensation. Component-related distortion can be dealt with by using additional parameters which are modified by each node in much the same way (depending on the number and type of components).

The difference between the two approaches for estimating BER is that OSNR and O-jitter are computed at each hop in the first approach, but only at the end point in the second approach. This makes the second approach much simpler computationally than the first. A disadvantage of the second approach is that more parameters are required in the **SETUP** message (as shown in Table 2).

Table 2: OSQ parameters required to estimate end-to-end BER; second approach.

OSQ Parameter	
n_0	Refractive index
$\partial n / \partial \omega$	First order wavelength dependence
$\partial^2 n / \partial \omega^2$	Second order wavelength dependence
$\partial^3 n / \partial \omega^3$	Third order wavelength dependence
$\partial n / \partial \lvert E \rvert^2$	First order intensity dependence
$\partial^2 n / (\partial \lvert E \rvert^2)^2$	Second order intensity dependence
L	Length traversed
--	Data parameters

E. Path Computation Component

The path computation component is responsible for constructing and updating burst forwarding tables at each node. Path computation is performed by a small number of *routing data nodes* (RDNs) in each domain. RDNs are responsible for **(1)** collecting data plane routing information for the domain, **(2)** computing a burst forwarding table for each node in the domain using OSQ parameters, and **(3)** distributing updated burst forwarding tables to each node in the domain.

One RDN is designated as primary for each domain, and others (if any) serve as backup RDNs to ensure that the network survives if the primary RDN fails. Backup RDNs perform all the functions of the primary RDN in background, and are ready to take over upon primary RDN failure. Nodes without RDNs are not involved in data plane path computation, and their routing databases contain no data plane information.

F. Forwarding Component

The forwarding component uses the burst forwarding tables stored at each node. The burst forwarding table contains information about the output interface for a burst, the output wavelength, the OSQ degradation expected on the output interface, the burst offset, and other information required to forward a burst. The availability of this information ensures that each OBS node can make a forwarding decision locally after consulting the OSQ fields of the **SETUP** message that precedes each burst's arrival. As noted, nodes also use OSQ parameters to decide whether a burst needs to be directed through a service circuit (e.g., gain, CD, PMD compensation.)

G. RDN Task 1 – Collecting Routing Information

RDNs are responsible for collecting data plane routing information for the domain. Each node monitors the status of its outgoing optical data interfaces and summarizes this information in an optical link state advertisement (OLSA). The information in an OLSA consists of all link attributes that are necessary for computing OSQ-guaranteed paths for the optical signal carrying the data bursts; e.g., the status of the interface (up, down), availability of optical resources (wavelengths, converters, splitters), and optical layer properties (impairments) that are relevant to routing. Each node transmits its OLSAs to the primary RDN in its domain via a point-to-point reliable connection over the signaling channel. The path to the RDN is determined by the control plane routing architecture, and is independent of the data paths that are computed by the RDN using OLSAs.

OBS nodes transmit OLSAs to the RDN whenever there is a change in the status of their data interfaces, or at specified intervals of time (in the absence of changes). Propagating

OLSA updates to the backup RDNs can be done in several ways – directly via separate point-to-point connections, or via a single multicast tree over the control plane, or by having the primary RDN periodically transmit all the OLSA updates it has received to each backup RDN via reliable point-to-point connections.

H. RDN Task 2 – Computing Burst Forwarding Tables

RDNs are also responsible for computing a burst forwarding table for each node in the domain using OSQ parameters. Once the RDN has collected OLSAs from each node in the domain, it uses a constraint-based routing and wavelength assignment (RWA) algorithm to compute data paths between each pair of nodes in its domain. The RWA algorithms use optical layer impairments to: **(1)** select a route – a sequence of physical links that guarantee a specified OSQ for the optical signal; **(2)** allocate the wavelength (or sequence of wavelengths) on which a burst will be transmitted; **(3)** compute the burst offset; and **(4)** detect and avoid routing loops.

1) Route Selection

Route selection requires choosing paths that guarantee a specific OSQ at the destination. The routing algorithm must consider both the physical layer impairments introduced at each link in the OBS network, and the OSQ (indirectly) requested by the application. (Recall that applications actually request QoS, and that requests are mapped to OSQ classes.)

We assume that the *magnitude* of physical layer impairments (loss, dispersion, etc.) on a given link is only a function of link (fiber) properties and the length of the link; and that *changes in magnitude* are a function of longer-term changes in link properties or characteristics (e.g., aging), and not a function of the instantaneous traffic (e.g., number of active wavelengths) carried by the link, which is stochastic. As noted, physical layer impairment information is part of the link state, and it is communicated to the RDN via OLSA updates that each OBS node transmits.

The RDN's routing algorithm is responsible for computing a set of alternate routes that meet the OSQ (rather than a single path) in order to ensure that the burst drop probability is low. The routing algorithm also considers the availability of compensation devices within the OBS network. Due to cost, it is expected that the number of compensation devices within the network will be small. Even so, the routing algorithm will likely be able to route a burst through a compensation device if it is determined that the optical signal would not otherwise achieve the requisite OSQ.

2) Wavelength Allocation

Wavelength allocation algorithms have been studied extensively within the context of circuit-switched WDM networks. These algorithms rely on information about the availability of wavelengths along the source-to-destination path. This information is valuable in circuit-switched environments, as the connection lifetime is several orders of magnitude longer than the time needed to collect the wavelength availability information. In OBS networks, burst transmissions are generally short-lived, and may be much shorter than the round-trip propagation time between source and destination. Consequently, any wavelength availability information advertised as part of the OLSA updates would have little value in selecting a wavelength, as this information is likely to be obsolete by the time the burst arrives at a downstream link.

Thus, we believe that information about the instantaneous occupancy (or availability) of wavelengths should not be included in OLSA updates that each OBS node periodically transmits to the RDN. Instead, we have devised new wavelength allocation policies that take into account statistical information regarding long-term wavelength usage at each link. Specifically, each OBS node is responsible for collecting statistics of wavelength usage for each of its interfaces. This information is collected by monitoring the status of the interfaces, and analyzing the feedback received by downstream nodes in the form of JIT **`FAILURE`** messages when a burst is dropped due to lack of a wavelength. The OBS node summarizes this information in the OLSA it sends to the RDN.

The RDN employs a wavelength selection algorithm to determine a set of possible wavelengths for each path. This set of wavelengths is part of the burst forwarding table that the RDN transmits to individual OBS nodes. Upon receiving a **`SETUP`** message, an OBS node uses the link and wavelength information in the burst forwarding table and the current state of its outgoing interfaces to determine the outgoing link and wavelength to be used for the incoming burst.

3) Burst Offset Calculation

RDNs must also provide estimates of the offset to be used for burst transmissions. We assume that the offset value is part of the burst forwarding table that the RDN provides to each OBS node. An ingress OBS node returns the offset value stored in its local forwarding table in the JIT **`SETUP ACK`** message to a client node in response to a **`SETUP`** message requesting permission to transmit a burst. If the burst forwarding table does not contain information for the destination requested by the client node, then the ingress node returns a pre-specified, long offset value.

Having an accurate offset estimate for each destination is important because a short offset may result in a dropped burst, while a long burst unnecessarily delays the burst. However, obtaining an accurate offset for each destination is difficult because the offset depends on the number of hops between the source and destination (which varies because alternate routing is used). It also depends on queuing delay that the **SETUP** message encounters along the signaling path (which varies based on congestion on the signaling path).

Hence, we use a feedback mechanism to estimate the offset value. *JumpStart* requires a downstream node that drops a burst to return a **FAILURE** message to the source with a reason for the dropped burst. The JIT protocol also specifies that the destination send a **CONNECT** message to the source. The information in a **FAILURE** (e.g., that the offset value was too short) or in a **CONNECT** (e.g., the overestimate of the offset at the destination) is used to adjust offset values higher or lower, respectively. This information is collected by the ingress node, summarized in an OLSA, and transmitted to the RDN. The RDN uses the information to adjust its offset value estimates. In a sense, offset values are dynamically adjusted to reflect the congestion along the burst paths.

4) Detecting and Avoiding Routing Loops

Transient routing loops are unavoidable due to the semi-centralized nature of the routing calculations for both data and control messages. Routing loops do not present a problem for control and management messages, and dealing with them is similar to dealing with routing loops in IP networks. The JIT frame format contains a time-to-live (TTL) information element similar to the TTL field of IP frames, and TTL is processed in JIT OBS networks in the same way as in IP networks. Decrementing TTL at each hop and discarding signaling messages whose TTL has reached zero will prevent signaling messages from cycling indefinitely, and will alert the network to the presence of a routing loop.

Data plane routing loops in a transparent optical network are far more serious. If the routing loop includes amplifiers, the signal may be amplified a number of times while looping and cause damage to the fiber plant or other network components.

To avoid data plane routing loops, we keep track of signal amplification levels via measurement and calculations. The **SETUP** message contains the power budget of the signal, and is updated when the signal passes through an amplifier. Signals whose power levels exceed recommended thresholds are discarded. Careful state maintenance at switches also allows the network to discover routing loops. If a **SETUP** message arrives at a switch for the second time, the message and related burst are discarded and the network is alerted.

I. RDN Task 3 – Distributing Burst Forwarding Tables

RDNs are also responsible for distributing updated burst forwarding tables to each node in the domain. Once the RDN has computed the data path, it converts the path information into burst forwarding tables for each node and transmits each table to its node via a reliable point-to-point connection. This transfer takes place over the control plane, using the shortest path between the RDN and the OBS nodes (computed by the link state routing protocol described in Section III). The semi-centralized route computation approach has several advantages:

1) Efficient Use of Computational Resources.

Routing algorithms that take into account constraints in the routing path (e.g., optical layer impairments) while attempting to optimize some performance objective are inherently expensive in terms of their computational requirements. With a semi-centralized routing approach, the network provider can concentrate the computing resources where they are needed (at the RDNs) rather than having to ensure that all OBS nodes have extraordinary computing power. RDNs can run complex, demanding algorithms to optimize the usage of the network resources.

2) Consistency of Routing Paths

Inconsistencies will arise if route computation is distributed and if each OBS node computes its own forwarding table independently of other nodes. (This is a common problem in IP networks.) The time required to resolve inconsistencies is related to the diameter of the network. For OBS networks running at 40 Gbit/s or beyond, inconsistencies that persist for a time interval equal to the network diameter will affect a significant amount of traffic. We believe that we can keep the length of time during which inconsistencies exist below a small threshold by using the semi-centralized approach described herein, and by using simple synchronization techniques.

3) Simplicity, Cost-Efficiency, and Upgradeability

Having a small number of nodes perform complex routing algorithms and protocols simplifies their design and reduces cost. A small number of nodes (i.e., the primary and backup RDNs) are involved if the routing algorithm is modified.

4) OLSA Distribution

Figure 4 shows the path computation component for the data plane routing architecture with the primary RDN (JITPAC-RDN) communicating with routing-aware JITPAC-Rs. As noted, point-to-point control plane paths are used to transmit OLSA information to the RDN, and to update the burst forwarding tables at each node. OLSA distribution takes place over control plane point-to-point paths. The burst routing database relies on OLSAs, but OLSA distribution relies only on the control plane routing database and not on the data plane routing database.

J. Summary – Intra-Domain Routing

The intra-domain routing protocols operate within a single administrative domain and perform three tasks:

(1) Bootstrapping. Bootstrapping allows the OBS network to self-configure upon restart, and allows nodes to join the network seamlessly. New nodes auto-discover their neighbors, and RDNs auto-discover new nodes and create /update/ distribute data and control forwarding tables.

(2) Link/node failure management. RDNs receive link state updates from JITPAC-Rs, recalculate routing tables upon failure, and distribute updated forwarding tables to some or all of the nodes in the domain.

(3) Distributing forwarding information. RDNs distribute and periodically update node-specific forwarding tables for both control and data traffic to each node.

Table 3 lists the main features of the intra-domain control plane and intra-domain data plane routing architectures.

V. Inter-Domain Control Plane Routing

Support for inter-domain routing of management and control messages is not required, as only data and data signaling cross domain boundaries.

VI. Inter-Domain Data Plane Routing

A. Approach

As noted, a JIT OBS network is a collection of transparent optical domains. Each domain is defined by its administrative control boundaries (company, campus, regional, economic, political). Architecturally, the network is represented by two orthogonal hierarchies – the *address* hierarchy and the *topological* hierarchy. In some instances the two may cleanly overlap, as when all nodes with the same address prefix are topologically located in the same domain. In other cases, the topology of the domains may not reflect their relationship in the address space. Disparate address spaces may be under the same administrative control, and thus the inter-domain routing system must be able to deal with such situations.

In order to route bursts to (or through) a domain, its neighbors must know the address prefixes reachable in the domain, and the minimum burst offset required to traverse a domain or deliver a burst within in. If the level of trust between domains is low, a domain may share no information other than a single worst-case offset sufficient to cover the entire domain. If the level of trust is high, a domain may share information (on a per-prefix basis) about OSQ and offsets that bursts should expect when entering or transiting the domain.

The routing architecture assumes complete transparency between the domains, either via optically transparent links between them or by providing conversion points for all available formats. There are security advantages to the latter approach, as it provides some assurance that no errant optical signal will enter the carrier's domain. Its disadvantage is the expense of providing converters for all data formats that the network may carry, and the likelihood that as optical technology matures, it will be possible to condition the optical signal without OEO conversion.

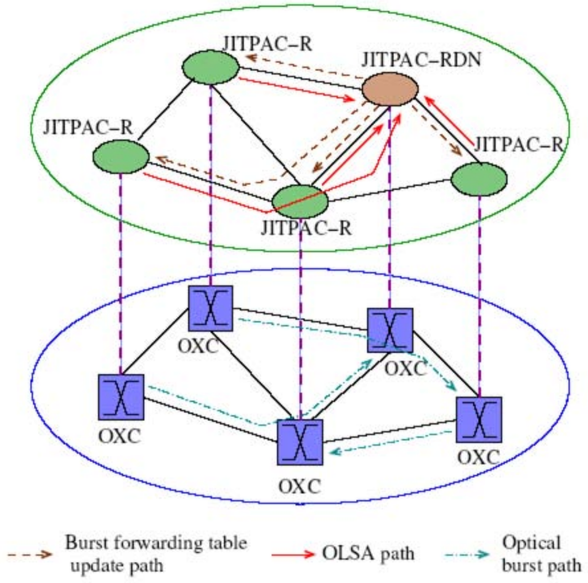

Figure 4: *JumpStart* intra-domain data plane routing

Table 3: Comparison of control and data plane intra-domain routing.

Feature	Control Plane	Data Plane
Objective	Shortest paths between JITPAC-Rs	OSQ paths between OXCs
Forwarding	Control plane forwarding table	Data plane (burst) forwarding table
Path computation	Fully or semi-distributed	Semi-centralized (at RDNs only)
Routing algorithm	Dijkstra's shortest path first	OSQ-based
Messages	LSAs	OLSAs and burst forwarding table updates
Message distribution via	Reliable, controlled flooding	Reliable, point-to-point connections
Implementation	Adapt existing protocols and algorithms	Design/develop/implement new protocols and algorithms

B. Architecture

Figure 5 shows three interconnected domains (A, B, C). Each of the links connecting two domains is terminated by *border switches* that belong to the respective domains, and that are administratively defined as having one or more links to neighboring domains.

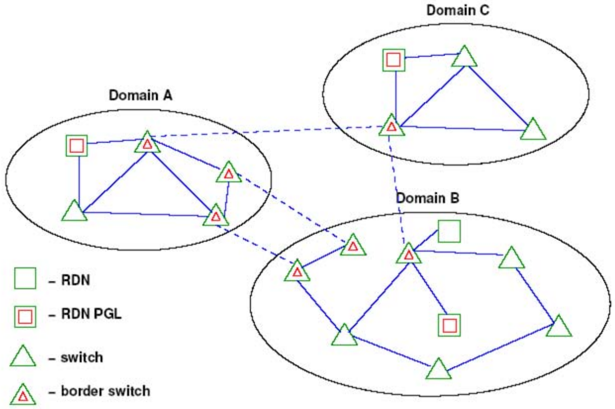

Figure 5: *JumpStart* inter-domain signaling architecture, using border switches and RDN peer group leaders (PGL).

RDNs are used to communicate with neighboring domains and to exchange routing information. The process includes the following stages:

(1) Neighbor domain discovery. Border switches discover and exchange information with peers in neighboring domains.

(2) Neighbor domain reporting. Border switches inform RDNs in their domain of inter-domain links, and the identity of the neighbor domains.

(3) RDN *peer group leader* (PGL) election. RDNs elect a PGL, which represents its domain to its neighboring domains.

(4) RDN PGL communications. RDN PGLs use a reliable transport protocol over the signaling infrastructure and a mix of source routing and normal control message forwarding. The RDN PGL source-routes its connection to the peer RDN PGL in another domain via its border switch; the border switch forwards the connection across the shared inter-domain link; and the peer border switch uses normal intra-domain routing to forward the connection to its RDN PGL.

RDN PGLs do <u>not</u> need to know the identity of peers in neighboring domains. This information is communicated to each border switch in the domain. When a border switch receives a request for an RDN-to-RDN connection, it forwards the request to its domain's RDN PGL. RDN PGL communications include authentication information and topological information (which depends on the policy-based level of trust established between domains). An RDN PGL may choose to share more information with some neighbors than with others.

C. JIT-TE for Inter-Domain Routing

The *JumpStart* architecture supports three resource provisioning schemes. (Table 4 summarizes the features of each.) "Traditional" JIT (JIT-OBS) is a tell-and-go scheme with fast but unreliable signaling – i.e., resource provisioning is not acknowledged link-by-link (which is the norm for most OBS architectures). JIT-OBS is well-suited for environments with very fast switching times (ns to low μs). Signaling is per-burst; i.e., each burst is preceded by its own **`SETUP`** message. JIT-OBS may be less desirable for longer-lived lightpaths in which the lightpath provisioning time is dwarfed by the holding time.

JIT with reliable signaling enhancements (JIT-RS) is a slower tell-and-wait scheme in which some JIT control messages are acknowledged link-by-link to ensure against loss. JIT-RS is well-suited for environments with slower switching times (low ms). Signaling is per-connection. Reliable signaling and end-to-end acknowledgements guarantee that a lightpath's resources will be provisioned prior to data transmission.

JIT with traffic engineering enhancements (JIT-TE) is an extension to *JumpStart* routing in which paths are established by the RDN in response to client requests. JIT-TE is also well-suited for environments with slower switching times, and for inter-domain routing. The connection **`SETUP`** request is passed from a node to the RDN using a reliable transport protocol. Unlike JIT-OBS and JIT-RS (which use JITPAC-R local forwarding tables to set up the route of the lightpath), JIT-TE relies on RDNs to build lightpaths by communicating with JITPAC-Rs in their domains and with RDNs in neighboring domains.

Domain-wide routing and direct communication between RDNs allows one to establish protection paths at the same time the primary path is provisioned. This approach precludes any requirement for a complex ATM-like crankback scheme.

D. Summary – Inter-Domain Routing

The inter-domain routing architecture operates across administrative domains. Its main task is to allow domains to learn of the presence of other domains, and to determine the best routes to ('destination') or through ('transit') these domains. The architecture is scalable because the number of domains may be quite large, and flexible in order to utilize policy-based routing to reflect both administrative and topological relationships between domains. The architecture also provides sufficient information to OBS nodes so that they can route data bursts across domains to meet minimal OSQ requirements.

Table 4: Comparison of three JIT resource provisioning variants.

Feature	JIT-OBS	JIT-RS	JIT-TE
Burst traffic	×		
Lightpath provisioning	×	×	×
Protected lightpath provisioning			×

Feature	JIT-OBS	JIT-RS	JIT-TE
Reliable signaling		×	×
Connection setup time	Low µs	Low ms	High ms
Maximum switching time	Ns to low µs	Low ms	Low ms
Offset	Small delay	One round trip time	More than one round trip time
Routing information provided by:	JITPAC-R local forwarding tables	JITPAC-R local forwarding tables	RDN domain-wide routing information base

VII. Conclusions

The *JumpStart* OEO control plane is responsible for conveying all signaling messages. Data bursts and their signaling messages must follow the same end-to-end path. Other signaling messages are not required to follow specific paths. The information required for routing bursts and burst control messages is very different from the information required for routing other control messages.

Hence, we have developed two routing implementations. This allows data-based signaling and other signaling to be optimized for their specific routing objectives. The data plane's objective is to compute paths that guarantee the OSQ of bursts between OBS endpoints. The control plane's objective is to compute paths between OBS nodes that support the efficient exchange of a variety of signaling messages.

We have developed architectures for intra-domain routing and for inter-domain routing. Intra-domain routing assumes that the domain is either optically transparent, or that it provides OEO conversion points to support all data formats. Inter-domain routing assumes that JIT signaling is terminated and reinitiated at domain boundaries.

We have developed a semi-centralized routing architecture for computing intra-domain paths. The architecture is able to estimate OSQ in several ways (hop-by-hop, end-to-end), and to map application QoS to a path's OSQ. We use a small number of semi-centralized RDNs in each domain to efficiently collect data plane routing information, and to compute and distribute burst forwarding tables to nodes.

We have developed a routing architecture for routing bursts to/through other domains. We use border switches that are administratively defined as having one or more links to neighboring domains, and RDN peer group leaders that represent their domains to neighboring domains. This allows various types of information to be exchanged between domains, depending on the level of trust.

We have developed two new provisioning schemes to augment per-burst JIT-OBS. JIT-RS is a reliable, tell-and-wait scheme in which control messages are acknowledged link-by-link, so signaling is per-connection rather than per-burst. JIT-TE allows reliable paths to established by the RDN in response to client requests, and is particularly well-suited for inter-domain routing.

References

[1] Baldine I., G. Rouskas, H. Perros, D. Stevenson, "JumpStart: a just-in-time signaling architecture for WDM burst-switched networks", *IEEE Communic.*, **40**(2), (2002).

[2] Zaim A., I. Baldine, M. Cassada, G. Rouskas, H. Perros, D. Stevenson, "Formal description of the JumpStart just-in-time signaling protocol using EFSM," *Proc. SPIE OptiComm 2002.*

[3] Zaim A., I. Baldine, M. Cassada, G. Rouskas, H. Perros, D. Stevenson, "JumpStart just-in-time signaling protocol: a formal description using extended finite state machines", *Optical Engineering*, **42**(2) (2003).

[4] Baldine I., G. Rouskas, H. Perros, D. Stevenson, "Signaling support for multicast and QoS within the JumpStart WDM burst switching architecture", *Optical Networks* **4**(6), November 2003.

[5] Baldine I., M. Cassada, A. Bragg, G. Karmous-Edwards, D. Stevenson, "Just-in-time optical burst switching implementation in the ATD*ne*t all-optical networking testbed", *Proc. Globecom 2003*, (IEEE, San Francisco CA, December 2003).

[6] Perlman R., *Interconnections*, Addison Wesley, Reading Massachusetts, 2000, Second Edition.

[7] Papadimitriou D. and D. Penninckx, "Physical Routing Impairments in Wavelength-switched Optical Networks," *Global Optical Communications*, 2002.

[8] Blumenthal D., "Performance Monitoring in Photonic Transport Networks," *Global Photonics Applications and Technology*, 2000.

[9] Strand J., A. Chiu, R. Tkach, "Issues for Routing in the Optical Layer," *IEEE Communic*., February 2001.

[10] ATM Forum, "PNNI Specification Version 1.0", af-pnni-0055.000, March 1996; "PNNI V1.0 Errata and PICS", af-pnni-0081.000, March 1997.

[11] Agrawal G., *Nonlinear Fiber Optics*, Academic Press, 2001.

[12] Ramaswamy R. and Kumar Sivarajan, *Optical Networks: A Practical Perspective*, Morgan Kaufmann Publishers Inc., 1998.

[13] Chraplyvy A., "High-Capacity Lightwave Transmission Experiments," *Bell Labs Technical Journal*, January 1999.

[14] Ronald Holzlohner R., V. Grigoryan, C. Menyuk, W. Kath, "Accurate Calculation of Eye Diagrams and Bit Error Rates in Optical Transmission Systems using Linearization," *Journal of Lightwave Technology*, March 2002.

[15] McKinstrie C., J. Santhanam, G. Agrawal, "Gordon-Haus timing jitter in dispersion-managed systems with lumped amplification," *J. Optical Society of America*, April 2002.