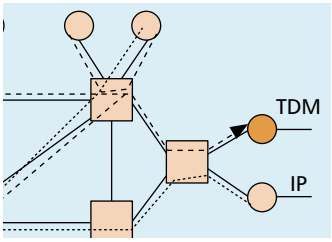

LIGHT CORE AND INTELLIGENT EDGE FOR A FLEXIBLE, THIN-LAYERED, AND COST-EFFECTIVE OPTICAL TRANSPORT NETWORK

INDRA WIDJAJA, IRAJ SANIEE, RANDY GILES, AND DEBASIS MITRA, BELL LABORATORIES, LUCENT TECH-

ABSTRACT

We present a new optics-based transport architecture that emulates fast switching in the network core via emerging fast tunable lasers at the network edge, and bypasses the need for fast optical switching and buffering. The new architecture is capable of handling both asynchronous and synchronous traffic, for dealing with various bandwidth granularities and responding to dynamic changes in end-to-end traffic demands. The architecture also reduces the amount of layering in the transport network by eliminating packet and TDM switching, keeps the network core light (lightweight and transparent), and pushes intelligence to the network edge. We discuss technical challenges that arise in the new architecture and describe possible approaches to address them.

INTRODUCTION

Next-generation transport networks must provide cost-effective transfer of disparate sets of client information, including multiservice traffic ranging from synchronous traffic (e.g., DS-1, DS-3, and STS-12) to asynchronous traffic (e.g., IP, Ethernet, and ATM). Such networks must also be flexible and responsive in dealing with different bandwidth granularities (e.g., from DS-1 to STS-192c), and dynamic changes in traffic demands (e.g., on the order of tens of milliseconds). Achieving these requirements simultaneously introduces new challenges to network designers, especially since the architecture cannot rigidly depend on the mix of clients or the bandwidth requirements as this information is likely to change in time.

Traditional transport networks are widely based on synchronous optical network (SONET) or synchronous digital hierarchy (SDH) rings [1]. Such networks rely on add-drop multiplexers (ADMs) and digital cross-connects (DXCs) to perform the switching, multiplexing, and demultiplexing functions of end-to-end connections. Traditional SONET-based networks are primarily designed to provide point-to-point connectivities for synchronous traffic, and are not well suited to supporting asynchronous traffic with a variety of bandwidth granularities or dynamic bandwidth requirements. Moreover, SONET rings do not scale gracefully as traffic demands in transport networks continue to rise.

Figure 1 illustrates two possible contrasting approaches for building the next-generation multiservice transport network. In the figure, switching — in wavelength, time-division multiplexing (TDM), or packet domain — takes place in the deeper shaded boxes, while adaptation and encapsulation take place at other boxes. Since switching is generally much more expensive than adaptation and encapsulation, a cost-effective network should seek a solution with minimal switching.

The circuit-centric approach adopts the evolution of a circuit-based transport network by

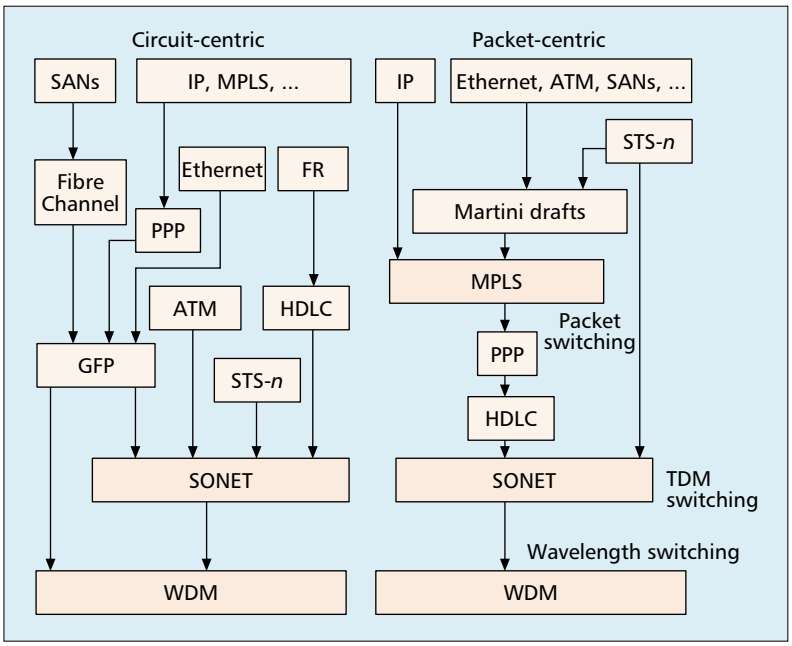

FIGURE 1. Candidate data-plane protocol stacks for multiservice transport networks.

0163-6804/03/$17.00 © 2003 IEEE

adding new capabilities such as bandwidth scaling through wavelength-division multiplexing (WDM) and support for flexible asynchronous traffic. Synchronous traffic and some asynchronous traffic (e.g., asynchronous transfer mode, ATM) may be transported directly over SONET through proper encapsulation and adaptation. Other types of asynchronous traffic (e.g., IP, Ethernet, storage area networks — SANs) can be encapsulated using a generic framing procedure (GFP), which promotes interoperability and provides a more efficient mapping method [2]. Recent extensions in the SONET specification facilitate a more flexible bandwidth granularity through the use of *virtual concatenation* and in-service bandwidth adjustment through the use of the *Link Capacity Adjustment Scheme (LCAS)* [3]. However, it is unlikely that LCAS can cope with a highly dynamic environment where bandwidth needs to be frequently adjusted, resulting in frequent setup and teardown of SONET connections and inducing stress on the signaling processor or management system (i.e., electronic/network management system, EMS/NMS). Another problem comes from the fact that TDM switching at the SONET level inherently requires large electrical cross-connects, which are expensive and wasteful if the switch ports already carry a substantial amount of traffic aggregation. Moreover, an electrical cross-connect requires an optical-electronic-optical (OEO) converter at each port, which tends to incur a relatively large portion of the system cost. Wavelength switching at the WDM layer may switch connections at coarse granularity cost-effectively. Optical cross-connects based on microelectromechanical system (MEMS) technology have been proposed as a promising wavelength switching candidate [4]. Optical cross-connects allow a network architecture that contains islands of transparency where optical signals are switched transparently (OOO) without the need for OEO conversion within an island. Due to the low per-unit cost of capacity for high-rate optical systems, transport network cost could be significantly reduced if the island boundary could be expanded closer to the clients. In reality, however, bandwidth requirements closer to the clients are typically small fractions of a wavelength capacity. Thus, in today's networks, electrical cross-connects are still needed, and the introduction of OOO would entail coexistence of two types of switching and result in additional network layering.

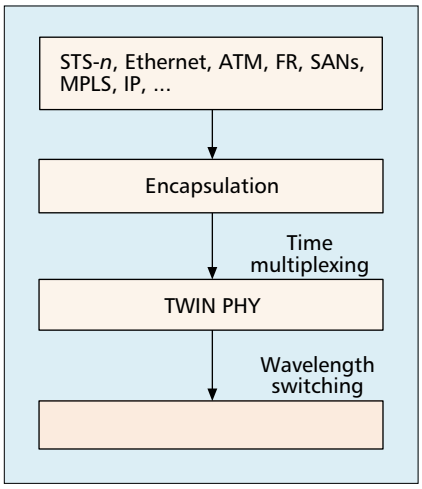

FIGURE 2. TWIN data-plane protocol stack.

The packet-centric approach adopts the evolution of a multiprotocol label switching (MPLS)-based network by adding new capabilities in adaptation and encapsulation mechanisms and extending the control plane (via general MPLS, GMPLS) to include other switching types (TDM and wavelength switching). This approach is based on the premise that the most widely adopted technology, argued to be MPLS in this case, deserves placement at the waist of the hour glass. It is well known that MPLS can easily transport layer 3 traffic such as IP. New encapsulation methods (so-called Martini drafts [5]) and extensions [6] have also been defined to transport various layer 2 (e.g., Ethernet and ATM) and layer 1 (e.g., SONET) traffic. Because packet switching is generally not cost effective for highly aggregated services, TDM and wavelength switching controlled by GMPLS signaling are generally needed to handle traffic at coarser granularities. Although this approach can handle both synchronous and asynchronous traffic, observe that three types of switching are required to achieve flexibility, thus making the overall solution unnecessarily expensive. Therefore, what is needed is not more network layering with switching at each layer and the extra complexities of adaptation, but a transport architecture with reduced network layers capable of handling both synchronous and asynchronous traffic well.

In this article we introduce a new approach called *Time-Domain Wavelength Interleaved Networking* (TWIN), which addresses the preceding requirements and is intended to be cost effective over a broad range of operating regions.

TIME-DOMAIN WAVELENGTH INTERLEAVED NETWORKING

Wavelength switching is indispensable as it is the basic building block for building future high-capacity scalable architectures. In TWIN, TDM switching and packet switching are emulated through the use of fast tunable lasers[1] capable of transmitting brief optical signals alternately at different wavelengths. Each ingress node of the transport network demultiplexes incoming data intended for different egress nodes into an outgoing optical signal with alternate wavelengths each associated with a particular egress node. In the network core, wavelength switching capable of routing optical signals ensures that each signal of a given wavelength arrives at the intended egress node. At the egress node, optical signals from various ingress nodes are multiplexed and converted back to electrical signals for further processing.

Figure 2 shows the TWIN data plane protocol stack with switching implemented only at the WDM layer. TWIN utilizes fast tunable lasers and wavelength switching to perform routing of various optical signals in the transport network. This architecture results in a single layer of switching and makes TWIN protocol stack simpler and cleaner than the preceding

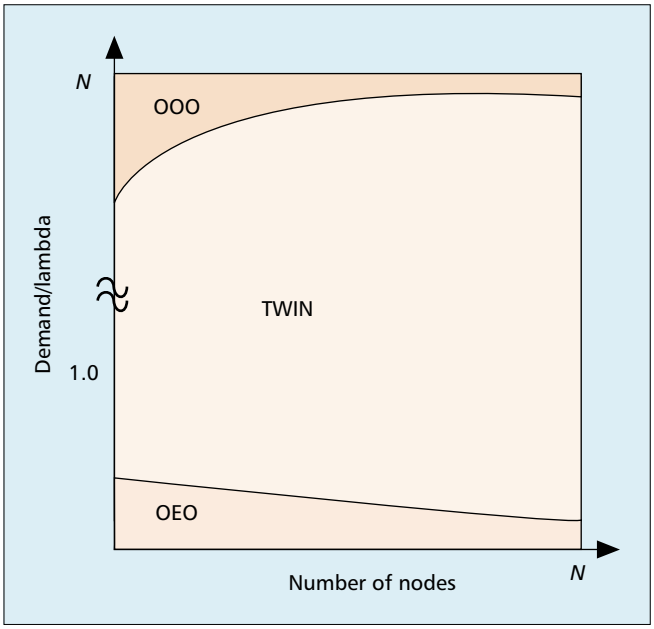

FIGURE 3 Comparison among OOO, OEO and TWIN.

[1] *A transmitter with a multifrequency laser can switch wavelengths in subnanoseconds [7].*

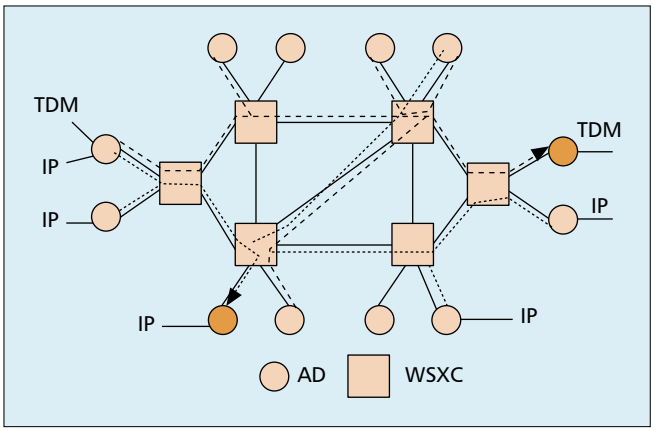

FIGURE 4. Logical multipoint-to-point trees, one for each of the two destinations, overlaid on top of a physical topology.

alternatives shown in Fig. 1. Figure 3 illustrates the general trade-off among OEO (TDM switching), OOO (wavelength switching), and TWIN as a function of the total traffic demand from each node (y-axis) and the number of nodes in the network (x-axis) under uniform traffic assumption (a particular trade-off depends on the particular set of costs). Each region indicates the technology that is most cost effective. At one extreme, when each node transmits close to the wavelength capacity to each other node, OOO is most cost effective as the network is already well utilized without grooming. At the other extreme, when the total demand from each node is a small fraction of the wavelength capacity, OEO is more cost effective due to its grooming capability. In most cases, when the demand is moderate to normal, TWIN generally presents the most cost-effective solution. Because of the unique architecture in TWIN, some new challenges need to be addressed to show its feasibility. We briefly outline some of these notable challenges and discuss them in more detail in the remainder of the article.

Network architecture: The general idea of using tunable lasers for switching in wideband optical networks has been documented in [8, 9]. Optical networks utilizing tunable lasers have only been applied to a simple physical topology such as a star or a bus. Support for an arbitrary topology, in our case, requires a switched network with wavelength-selective cross-connects capable of merging incoming signals of the same wavelength to the same outgoing fiber. Our approach enables network resources to be allocated in an automated and flexible fashion to match traffic demand and other conditions. Support for an arbitrary topology presents new challenges in scheduling, protection, routing, and signaling, which have not been addressed in this context in the past.

Framing and encapsulation: TWIN must implement burst-mode receivers to perform frequency synchronization to recover the transmitter's clock rapidly, provide framing to determine the boundary of a data unit, and encapsulate client protocol data units (PDUs).

Scheduling: Scheduling traffic between each input-output pair in a single switch is a well studied problem. In TWIN, however, the scheduling algorithm must deal with the substantial delays due to propagation of signals across the network.

Protection: The transport network is expected to be reliable in that traffic interruption due to a fault in the network should be restored rapidly. TWIN requires redundancy in multipoint-to-point trees.

Control plane: TWIN adopts a separate control plane to perform automatic discovery of resources, routing, and signaling. Routing in TWIN is unique since a bandwidth bottleneck can only occur at the source/destination. The light intermediate network elements in TWIN also give rise to a new signaling model.

NETWORK ARCHITECTURE

The transport network consists of a *wavelength-selective cross-connect* (WSXC) in the network core and *aggregation devices* (ADs) at the edge, as shown in Fig. 4. To provide effective transport connectivities among the ADs, TWIN makes use of *optical multipoint-to-point trees* that are overlaid on top of the physical topology. Unlike multipoint-to-point trees in packet switching (e.g., ATM or MPLS) that are solely intended to minimize labels, the main motivation for using trees in TWIN is to maximize the utilization of the transmitter/receiver. Figure 4 provides an example where such trees are associated with two destinations, and each destination is assigned to a unique wavelength. Thus, in general, a network with N ADs would need $O(N)$ wavelengths.[2] This requirement is to be contrasted with a pure OOO solution (with no electronic grooming), which provides a point-to-point wavelength connectivity between each pair of nodes, resulting in a total of $O(N^2)$ wavelengths. In TWIN, sources that have data to trans-

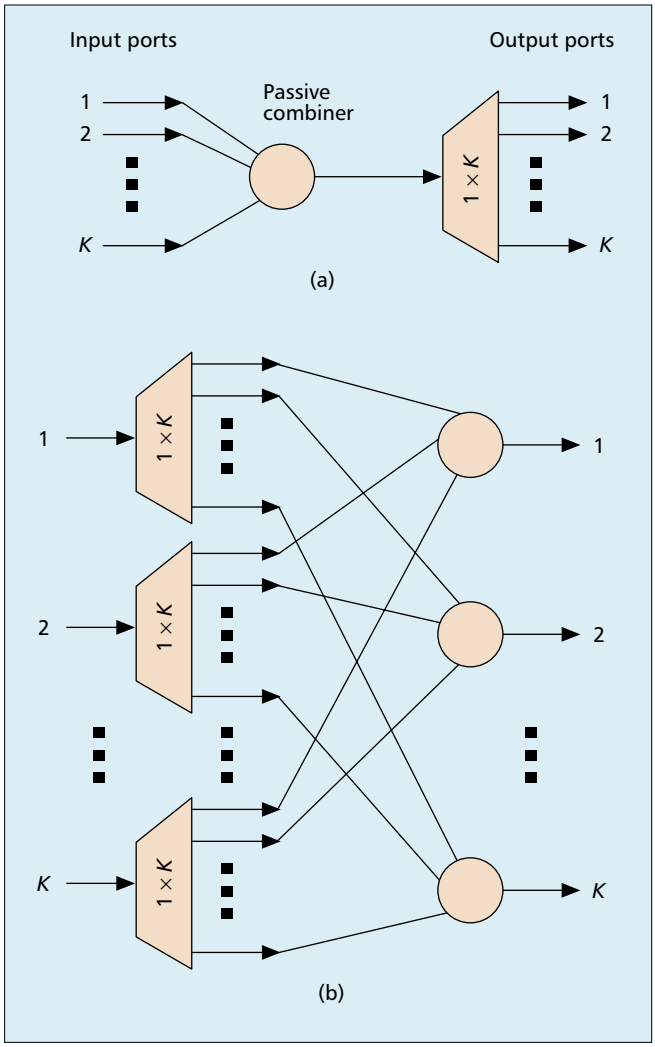

FIGURE 5. Wavelength-selective cross-connect with merging: a) full merging; b) partial merging.

[2] *If the total bandwidth for a given destination exceeds the wavelength capacity, multiple wavelengths need to be assigned to that destination.*

mit to a particular destination use the wavelength assigned to that destination. The optical signals from various sources to a particular destination may be merged at the intermediate nodes. Thus, sources must coordinate their lasers so that collision will not occur at each merging point (or at the destination). TWIN relies on fast tunable lasers and scheduling to prevent such collisions (discussed later).

Each AD functionally consists of ingress AD (source) and egress AD (destination). The ingress AD aggregates incoming client traffic flows for each egress AD, encapsulates client PDUs (e.g., IP packets, ATM cells, MPLS frames) for the same egress AD into the same TWIN PDUs (or *bursts*[3]), and optically transmits each burst using a fast tunable laser. The wavelength assigned to each burst is used to route the burst to its intended destination. The egress AD demodulates the received optical signal, decapsulates the burst into individual client PDUs, and forwards these PDUs to the appropriate client's ports.

The WSXC performs self-routing of incoming optical signals to the appropriate outgoing fibers based on the wavelengths of the signals. In contrast to the traditional optical cross-connect, TWIN requires the WSXC to merge incoming signals of the same wavelength to the same outgoing wavelength. An example of such a WSXC is shown in Fig. 5a where full merging is performed by a passive combiner at the ingress side. The $1 \times K$ switch routes each individual wavelength to the appropriate output port (fiber) [10]. In certain cases (e.g., when wavelength reuse is critical), input signals of the same wavelength may need to be routed to different output ports. Figure 5b shows a WSXC capable of separating such signals to different output ports. Two important observations are worth pointing out. First, note that a merging WSXC can be significantly simpler than a typical WSXC as a merging WSXC may be implemented by a $1 \times K$ switch, as shown in Fig. 5a. Second, unlike other approaches that require fast reconfigurable (every hundreds of nanoseconds to microseconds) cross-connects at the burst level [11], TWIN purposely avoids fast reconfigurability in the core since our cross-connect reconfiguration is only needed when a new connection requires a new branch of a tree to be created (typically on the order of minutes or even much longer once the trees have been constructed). TWIN relies on fast tunable lasers at the network edge to emulate fast reconfigurability in the network core.

Framing and Encapsulation

TWIN requires a burst mode receiver at each destination since the receiver needs to handle bursts from different transmitters that are clocked asynchronously. Frequency synchronization for a burst mode receiver can be facilitated by a preamble in each burst so that the receiver's synchronizer can lock to the transmitter's bitstream. Once bit synchronization is achieved, burst delineation can be accomplished by appending a start-of-burst delimiter field. A burst mode receiver capable of performing this synchronization within 50 ns has been successfully demonstrated [12].

The TWIN burst format adopts the GFP specification [13]. However, changes are necessary to accommodate features unique to TWIN. Unlike GFP, which uses cyclic redundancy check (CRC)-based framing, TWIN relies on framing based on preamble and start-of-burst delimiter due to the asynchronous nature of the transmitter-receiver pair. GFP payload length field, which is 2 bytes long, can be used to accommodate payload information of up to 64 kbytes length. This limits the duration of a burst to about 52 μs with 10 Gb/s transmission rate or about 13 μs with 40 Gb/s transmission rate For flexibility, an extension of the payload length field is needed. Furthermore, TWIN generally maps multiple client PDUs into a burst. To do so, a PDU length field has been added to delineate each client PDU within the payload. GFP encapsulation appears flexible enough to handle future extensions that are provided in the payload header via optional fields for extensions.

[3] *A typical length of a burst is explained later.*

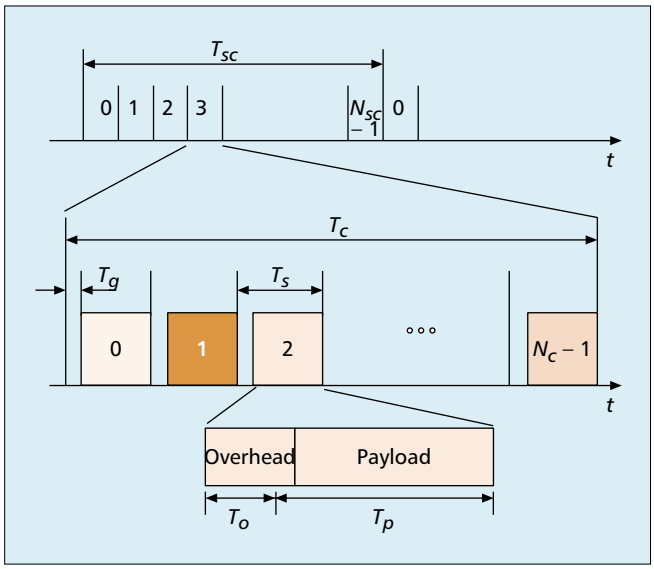

FIGURE 6. Scheduling cycles.

Scheduling

Each multipoint-to-point tree can be viewed as a shared medium associated with a given destination. Two basic approaches for accessing a shared medium are random access and scheduling. Random access is not suitable in the context of TWIN where propagation delays may be significant and transmission rate is very high. Thus, TWIN adopts the scheduling approach.

Transmission of typical synchronous traffic to a given destination is organized in repetitive *cycles* of duration T_c each. A cycle consists of N_c slots, each of duration T_s. A slot carries exactly one burst. Adjacent bursts are interspaced by a *guard time* of duration T_g. Each burst consists of an overhead of duration T_o and payload of duration $T_p = T_s - T_g - T_o$. Suppose TDM frames (e.g., SONET frames) with smallest granularity of size F bits are to be transmitted periodically in one burst per cycle. Then the number of TDM frames that can be placed into one slot is $N_f = T_p R/F$, where R is the optical transmission rate. Since a TDM frame needs to be repeated every 125 μs, the slot interval for this channel, which is equivalent to T_c, is $125 \times N_f$ μs. Figure 6 shows the relationship among different parameters. To handle client traffic with low transmission rate, we can extend the periodicity by adding a *supercycle* so that such traffic is transmitted periodically every T_{sc}.

Table 1 shows an example of how the slot size (T_s) affects various parameters such as the frame duration (T_c), efficiency (η), and buffer requirement (buffer) for F = 6480

T_s (ms)	N_f	T_c (μs)	N_c	η (%)	Buffer (kB)
2	2.70	337.58	168	87.1	717
4	5.86	733.03	183	94.9	1697
6	8.95	1118.83	186	96.4	2633
10	15.12	1890.43	189	97.9	4521
20	30.55	3819.45	190	98.5	9184

TABLE 1. Parameters for STS-1 granularity (kB: kbytes).

Let d_{ij} be the fixed propagation delay between source i and destination j. Let $S_{ij}(k)$ be one when a burst is sent at slot k from source i to destination j, zero otherwise, and $D = [D_{ij}]$. TIIS performs as follows:

1. Set $S_{ij}(k) \leftarrow 0$, $k = 0, \ldots, T-1$

2. Set $\hat{D} \leftarrow D$

3. Let $(i^*, j^*, k^*) = \arg\max_{i,j,k} \dfrac{f_{ik}(k)\hat{D}_{ij}}{\sum_{i',j',k' \in \Gamma_{ij}(k)} f_{i',j'}(k')\hat{D}_{i'j'}}$

4. Set $S_{i^*j^*}(k^*) \leftarrow 1$, $\hat{D}_{i^*j^*} \leftarrow \hat{D}_{i^*j^*} - 1$

5. Repeat (3) until either $f_{ij}(k) = 0$ for all (i, j, k) or $\hat{D} = 0$, where a feasible burst transmission for (i, j) at slot k is $f_{ij}(k) = (1 - \Sigma_j S_{ij}(k))(1 - \Sigma_{i'} S_{i'j}((k + d_{ij} - d_{i'j}) \bmod T))$, and neighbors of edge (i, j) at slot k, $(\{i, j, k\})$, is $\Gamma_{ij}(k) = \{i, j', k; \forall j'\} \cup \{i', j, k' : (k' + d_{i'j}) \bmod T = (k + d_{ij}) \bmod T\}$.

bits (STS-1), $T_g = 200$ ns, $T_o = 50$ ns, and $R = 10$ Gb/s.

The purpose of scheduling is to assign appropriate slot(s) to source-destination pairs to ensure that collisions do not occur, maximize slot utilization, and ensure a minimum rate for each node pair. To support both synchronous and asynchronous traffic, TWIN adopts both a centralized scheduler (CS) and distributed scheduler (DS).

The CS matches the behavior of synchronous traffic, where client information frames arrive periodically (every 125 μs) and the bandwidth of a connection is relatively fixed. Since the CS can gather all necessary information (e.g., traffic demand matrix) and process this information in a relatively long time interval (on the order of seconds), the CS can compute the slot allocations to each source-destination pair very effectively. Unlike the problem of scheduling in a packet switch, the TWIN scheduler must estimate and incorporate the scheduling delay information to deal with various propagation delays that exist between various nodes. The scheduling information computed by the CS is downloaded to each AD through a data communications network (DCN). The DS is suitable for asynchronous traffic with dynamic bandwidth requirements. For faster response time, each DS is associated with a destination and performs scheduling among the sources wtih information to send to that destination. The DS assigns scheduling slot(s) to a source by examining a request sent by a source and granting certain slots in subsequent (super) cycles. The request and grant messages are communicated in-band for fast transmission so that changes in request can be reflected quickly. To ensure that the DS and CS do not allocate the same slots, each cycle is divided into a *sync period* for transmission of synchronous traffic and an *async period* for transmission of asynchronous traffic. The boundary of the two periods is flexible and can be negotiated between each CS and DS pair.

We now consider the CS in more detail. The problem of scheduling with no propagation delay can be modeled as a bipartite graph with weighted edges connecting each source i on the left and each destination j on the right. The edges represent the traffic demands. Given the traffic demand D_{ij} from each source i to each destination j in units of time slots per period, the lower bound for the number of slots required to schedule all the demands is $T^* = \max\{\max_j \Sigma_i D_{ij}, \max_i \Sigma_j D_{ij}$. In situations where propagation delays are either all equal or negligible, this lower bound is also sufficient. In general, this lower bound will be exceeded when the propagation delays are arbitrary. We present the *TWIN Iterative Independent Set (TIIS)* algorithm whose basic idea is to find the maximum-weight independent set of edges.[4] This is done iteratively where in each iteration the largest set (in terms of total weight) of "non-neigboring" edges is selected. The set of edges and weights are updated until there is no more demand to schedule. Propagation delays need to be accounted for appropriately to ensure that a selected set of edges is feasible (i.e., independent or non-neigboring). To determine a schedule and check feasibility of a given number of slots T, the TIIS algorithm performs as described in the box.

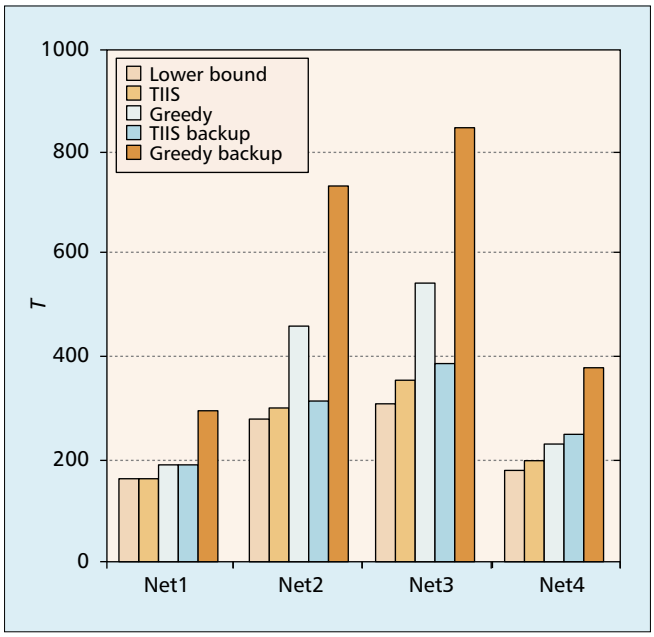

FIGURE 7. Scheduling with propagation delays.

Figure 7 provides a comparison between TIIS and a greedy algorithm to schedule synchronous traffic in four different real network scenarios outlined in Table 2 (real names have been replaced with network numbers). The results indicate that TIIS is typically within 10 percent of the lower bound T^*, while a greedy algorithm that blindly selects the highest demand may need 50 percent more slots. TWIN requires time-of-day synchronization to ensure that each source uses a common time reference for assigning bursts into slots. TWIN relies on a GPS-based timing reference to provide a low-cost and accurate time-of-day information. Receivers capable of maintaining time-of-day accuracy within 100 ns to GPS and satisfying telco requirements (i.e., NEBS compliance) are available in the marketplace.

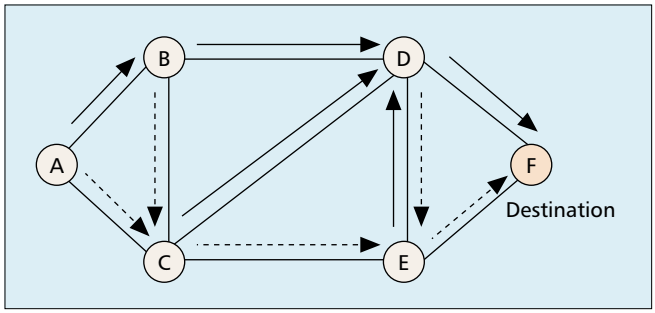

FIGURE 8. Redundancy with multitrees for a given destination node F (the AD and WSXC are assumed to be integrated)..

[4] *More details on TIIS are described in [14].*

PROTECTION

Transport networks must be survivable so that a failure in the network will only cause disruption locally and for a short period of time. Survivability may be provided through path protection. In 1+1 path protection, traffic is copied on both the *working* and *protection* paths, while the destination performs the selection between the two paths based on some criteria (e.g., alarm indication signal or bit error rate). The restoration time for 1+1 protection is generally very fast because failure localization and notification are not required to perform protection switching. However, 1+1 protection has a disadvantage in that the bandwidth (slot) requirement is doubled.

In 1:1 path protection, traffic is sent to the working path during normal operation. When a failure occurs on the working path, traffic is automatically switched to the protection path. 1:1 protection is more bandwidth efficient than 1+1 protection. However, 1:1 protection is not as responsive as 1+1 protection, since a failure needs to be identified at the destination, and then the associated notification message needs to be propagated to the affected sources. Unlike typical protection, we need to consider the impact of TWIN protection on scheduling efficiency. A joint schedule with 1:1 protection can be constructed by sharing time slots on the working and protection paths to increase efficiency. Figure 7 shows the effect of joint scheduling of working and protection paths ("TIIS backup") on the number of time slots required. As can be seen, the penalty for joint scheduling is relatively small compared to the case with no protection ("TIIS").

	Net1	Net2	Net3	Net4
Nodes	10	11	6	10
Links	16	16	15	16
Distances	150–2000	50–600	1–20	100–2000
Demand pattern	Uniform	Nonuniform	Highly nonuniform	Close to uniform

TABLE 2. Network parameters for Fig. 7.

Path design in TWIN protection presents a new twist since working and protection paths need to be part of multipoint-to-point trees. Tree redundancy can be created using edge-disjoint trees, arc-disjoint trees, or multitrees [15]. Figure 8 shows an example of multitrees using the ear decomposition approach described in [15]. The dashed and solid trees to destination F are constructed such that each source maintains its connectivity to the destination for any link or node failure (other than the destination).

CONTROL PLANE

The main tasks of a control plane are resource discovery, routing, and signaling. Automatic neighbor discovery to discover and maintain the link between neighbors can be designed by appropriately modifying the commonly used Hello protocol. Link management tasks may follow the approach specified in Link Management Protocol (LMP). Network discovery pertains to gathering of data plane topology and resource information in the entire network. A reliable flooding protocol such as Open Shortest Path First (OSPF) can be adopted for network discovery.

Routing in TWIN is unique in two interesting aspects. First, unlike traditional transport networks that may optimize path selection based on the available bandwidth on each link, the bandwidth bottleneck in TWIN only occurs at the source/destination. Thus, choosing different intermediate links generally will not affect resource consumption. Second, the need for redundant multitrees to support survivability implies that routes for each destination need to be precomputed. When a source wishes to establish a connection to a destination, the source has to follow the route given by the multitrees associated with that destination. These two aspects imply that a centralized route server presents a compelling case for a simple and efficient routing implementation.

A signaling protocol is used to set up, maintain, modify, and tear down connections. Currently, GMPLS protocols, such as Resource Reservation

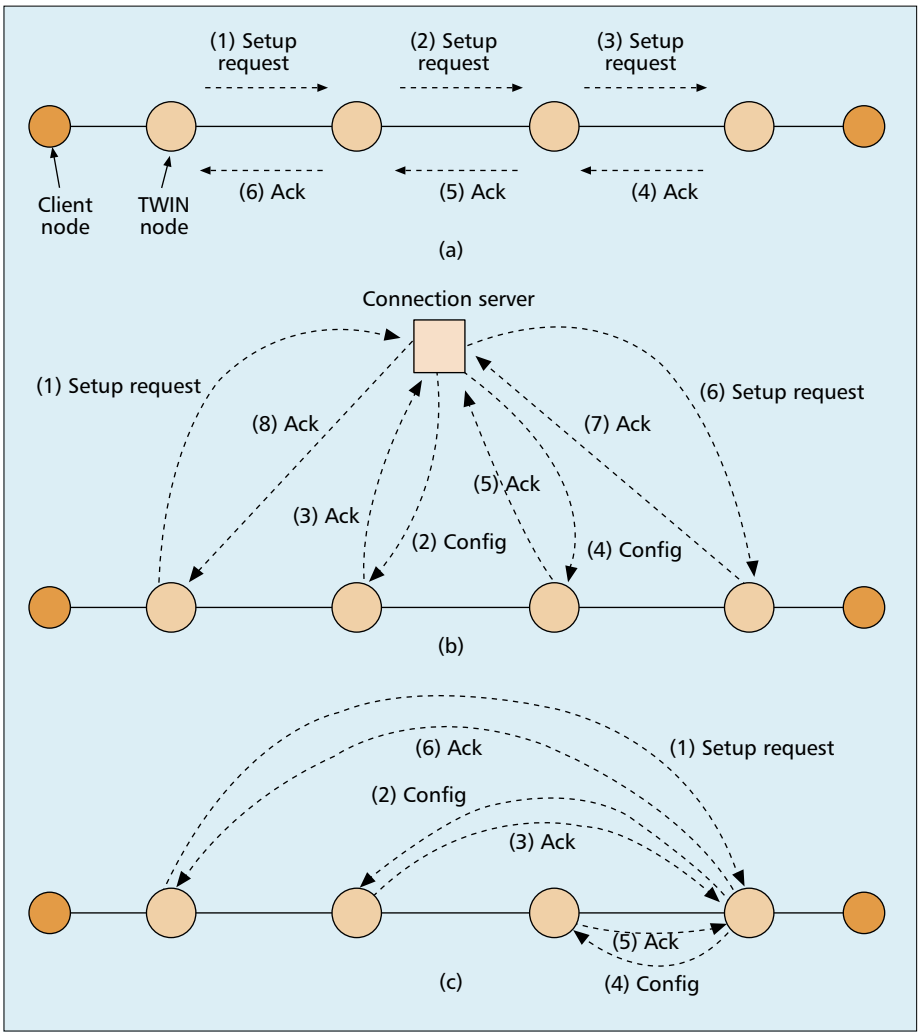

FIGURE 9. Signaling model: a) hop-by-hop model; b) rendezvous model; c) end-to-end model.

Protocol with Traffic Engineering (RSVP-TE), appear to be the most widely followed approach to transport network signaling. The propagation of signaling messages in RSVP-TE is similar to that in many other signaling protocols whereby a *setup request* (implemented using the Path message in RSVP-TE) is propagated from the source hop by hop to the destination, as shown in Fig. 9a. If the request is granted by the destination, an acknowledgment (ack), implemented using the Resv message in RSVP-TE, is propagated hop by hop in the reverse direction. The ack contains the wavelength and slot number(s) to be used by the source. Unfortunately, this *hop-by-hop model* is generally not efficient for TWIN. The reasons are twofold. First, unlike traditional transport networks that have to configure a unique incoming label to a unique outgoing label for a new connection at each node, TWIN uses the same label (wavelength) assigned to each destination. This implies that TWIN rarely needs to configure the cross-connects along the path. Second, unlike traditional transport networks that have to perform connection admission control (CAC) at each node, TWIN only has to perform CAC at the source and destination since the bottleneck can only occur at the source or the root of a multipoint-to-point tree. Thus, a more efficient signaling mechanism taking these two aspects into consideration is needed.

An alternative signaling model, called the *rendezvous model*, is illustrated in Fig. 9b. Here the source (node A) first sends a setup request to a well-known connection server. Upon receipt of a setup request, the connection server configures the intermediate cross-connects (at nodes B and C), if not already done, by sending appropriate configuration messages. The connection server then sends a setup request to the destination (node D). If the request is accepted, the destination returns an ack to the server, which in turn replies to the source. Note that steps 2–5 are skipped when the multipoint-to-point tree spanning nodes B and C has been built, thereby resulting in fast signaling time. The rendezvous model is attractive when routing and scheduling are also performed at the server where the state of all trees is known. A third signaling model, called the *end-to-end model*, is illustrated in Fig. 9c. In this model, the source sends the setup request directly to the destination. If necessary, the destination would first configure the intermediate cross-connects. Finally, the destination checks its resource availability, and, if available, returns an ack to the source. The end-to-end model requires each destination to maintain the state of the multipoint-to-point trees rooted to it. Once the trees have been built, the end-to-end model requires only two exchanges of signaling messages. Finally, we note that TWIN signaling manages only wavelength connections, thus removing the complexity of hierarchical connections in networks having multiple (i.e., packet, TDM, and wavelength) layers.

Conclusion

The current proposed transport network architectures based on circuit- and packet-centric approaches are not cost effective and are complicated in layering. A cost-effective and flexible approach is to reduce the amount of layering in the transport network, keep the core light, and distribute the intelligence to the edge. By exchanging fast switching in the network core with fast tunable lasers at the network edge, we render the network able to function effectively like a switch whose input/output ports are spread across the network nodes. We present a new architecture that addresses key requirements for providing a flexible, cost-effective, multiservice solution. We also discuss new technical challenges that arise in this unique architecture, and describe possible approaches.

Acknowledgment

The authors thank Martin Zirngibl for sharing his expertise on fast tunable lasers and the anonymous reviewers for their useful comments.

References

[1] ANSI T1.105, "Synchronous Optical Network (SONET)," 1995.
[2] E. Hernandez-Valencia, M. Scholten, and Z. Zhu, "The Generic Framing Procedure (GFP): An Overview," *IEEE Commun. Mag.*, May 2002, pp. 63–71.
[3] P. Bonenfant and A. Rodriquez-Moral, "Generic Framing Procedure (GFP): The Catalyst for Efficient Data over Transport," *IEEE Commun. Mag.*, May 2002, pp. 72–79.
[4] D. J. Bishop, C. R. Giles, and G. P. Austin, "The Lucent LambdaRouter: MEMS Technology of the Future Here Today," *IEEE Commun. Mag.*, Mar. 2002, pp. 75–79.
[5] L. Martini *et al.*, "Encapsulation Methods for Transport of Layer 2 Frames over IP/MPLS Networks," IETF, work in progress, Feb. 2003.
[6] A. Malis *et al.*, "SONET/SDH Circuit Emulation over Packet (CEP)," IETF work in progress, Jan. 2003.
[7] M. Kauer *et al.*, "16-Channel Digitally Tunable Packet Switching Transmitter with Sub-Nanosecond Switching Time," *Proc. ECOC*, paper 3.3.3, 2002.
[8] B. Mukherjee, WDM-Based Local Lightwave Networks Part I: Single-Hop Systems," *IEEE Network*, May 1992, pp. 12–26.
[9] S. Alexander *et al.*, "A Precompetitive Consurtium on Wide-Band All-Optical Networks," *J. Lightwave Tech.*, vol. 2, no. 5/6, May/June 1993.
[10] D. M. Marom, "Wavelength Selective 1 × K Switches for Transparent Optical Networks," *Proc. SPIE*, vol. 4907, Oct. 2002.
[11] C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration," *IEEE Commun. Mag.*, Sept. 2001, pp. 104–14.
[12] Y. Su *et al.*, "Demonstration of a WDM-TDM Metropolitan Ring Network Prototype," Bell Labs tech. memo, Aug. 2002.
[13] ITU-T Rec. G.7041, "Generic Framing Procedure (GFP)," Dec. 2001.
[14] K. Ross *et al.*, "Scheduling Bursts in Time-Domain Wavelength Interleaved Networks," submitted for publication, 2003.
[15] M. Médard *et al.*, "Redundant Trees for Preplanned Recovery in Arbitrary Vertex-Redundant or Edge-Redundant Graphs," *IEEE/ACM Trans. Net.*, vol. 7, no. 5, Oct. 1999.

Biographies

INDRA WIDJAJA received a B.S. degree from the University of British Columbia, an M.S. from Columbia University, and a Ph.D. from the University of Toronto, all in electrical engineering. From 1994 to 1997 he was an assistant professor of electrical and computer engineering at the University of Arizona. From 1997 to 2001 he was with Fujitsu Network Communications. Since 2001 he has been a research member at Bell Laboratories. He is the co-author, with Leon-Garcia, of the textbook *Communication Networks: Fundamental Concepts and Key Architectures* (McGraw-Hill).

IRAJ SANIEE is head of the Mathematics of Networks and Systems Research Department at Bell Laboratories, Lucent Technologies. His recent research has been in optimization of resource sharing optical networks, development of algorithms for network design models, and performance evaluation for multiscaling models of data traffic. He has published over 40 articles in IEEE, SIAM, and INFORMS journals and proceedings. He received his B.A., M.A. and Ph.D. in operations research and control theory from Cambridge University.

RANDY GILES is director of the Advanced Photonics Research Department at Lucent Technologies. Research programs in his department include the study of new optical materials, characterization and utilization of light in optical communications, and development of optical networking technologies. In his 16-year career at Bell Laboratories, he pioneered the modeling and use of erbium-doped fiber amplifiers for lightwave systems, demonstrated the first optical ADMs by means of Bragg grating technology, and developed optical network applications of micromachines including scalable optical crossconnects and ADMs. He is a graduate of the Universities of Alberta and Victoria. Before Bell Laboratories, he worked at Nortel's research labs on their first gigabit optical transmission systems. He is a fellow of the Optical Society of America and was awarded the 2001 Bell Laboratories Fellowship.

DEBASIS MITRA [M] is vice president of mathematical sciences research at Bell Laboratories. He received a Ph.D. degree in electrical engineering from London University and joined Bell Laboratories as a member of technical staff in 1968. During the fall semester of 1984 he was Visiting McKay Professor at the University of California, Berkeley. His current interests are in optical networking, IP/optical convergence, stochastic traffic engineering, network economics and network revenue management. He is a member of the National Academy of Engineering and a Bell Laboratories Fellow. He is the recipient of awards given by the Institution of Electrical Engineers (United Kingdom), *Bell System Technical Journal*, the 1995 ACM Sigmetrics/Performance Conference, the 1993 Steven O. Rice Prize Paper Award, and the 1982 Guillemin-Cauer Prize Paper Award of the IEEE. He is a co-recipient of the 1998 IEEE Eric E. Sumner Award with the following citation: "For the

conception and development of voice echo cancelers." He has been a member of the editorial boards of *IEEE/ACM Transactions on Networking*, *IEEE Transactions of Communications*, *IEEE Transactions on Circuits and Systems*, and *Queuing Systems* (QUESTA). He is currently area editor of Operations Research for *Telecommunications and Networking*.