

A New Transport Network Architecture with Joint Time and Wavelength Multiplexing

Indra Widjaja and Iraj Saniee
Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07974, USA

Abstract—Current transport network architectures are complex and expensive to manage as they typically consist of multiple layers (packet, TDM and WDM layers). We propose a cost-effective transport network architecture that relies on a single data-plane layer by exploiting joint time and wavelength multiplexing at the WDM layer. We describe the architecture that emulates fast switching in the network core with fast tunable lasers at the network edge, and enables self-routing of optical signals through passive wavelength-selective cross-connects. We present some results on distributed scheduling that arbitrates the transfers of optical signals from various sources to various destinations so that conflicts are avoided.

I. INTRODUCTION

While the optical bandwidth in the network core has been scaling to very high capacity due to advances in WDM technology, end-to-end traffic demands typically only require bandwidth of much smaller magnitude. Traditional multi-layer networks help bridge this gap through traffic grooming which allows different end-to-end traffic streams to be multiplexed and demultiplexed at various nodes in a network. The emerging network mechanisms such as GMPLS simplify control plane and facilitate grooming in a multi-layer network through unified routing and signaling protocols [1]. The main benefit of using GMPLS is the reduction of service provisioning cost through automated setups and teardowns of connections. However, these networks still require switching at each layer (e.g., packet, TDM and WDM layers) and intelligence within the network core. Furthermore, other network functions such as protection, traffic engineering and network management still need to be performed at each layer.

Given that the network cost is dominated by the number of layers that has to be managed, these new network control mechanisms will not significantly reduce the cost of operating the network. In this paper, we propose a disruptive network architecture that relies on a single (WDM) layer in the data path within the network core. Intelligence that deals with various data processing functions resides exclusively at the network edge. The resulting architecture has several desirable properties. First, simplification in the network core significantly reduces the cost of “transit ports”, which are used to simply forward traffic in the network core. Second, the “service ports” which provide direct service interfaces to the clients (e.g., IP routers, ATM switches and SONET ADMs), reside only at the network edge. The cost advantage of this architecture can be even magnified as service ports need only be deployed incrementally as the number of clients grows. Moreover, as the ratio of the number of transit ports to service ports is proportional to the average number of hops, it becomes even more desirable to optimize the cost of transit ports.

By reducing the network core to a single layer, key network

functions such as protection and traffic engineering only need to be implemented in one layer. This is in contrast to a multi-layer network (e.g., GMPLS-based network) where these network functions are essentially duplicated at each layer. Moreover, complicated inter-layer correlation, escalation or information exchange do not arise in a single-layer network. Finally, the proposed architecture also promises to significantly simplify network management which typically accounts for a significant cost of network operation.

To provide arbitrary bandwidth for end-to-end applications, the proposed architecture leverages joint time- and wavelength-division multiplexing. We review the salient features of the proposed architecture and highlight the new networking problems that still need to be solved. We provide preliminary results of one key component of the proposed architecture, namely, distributed scheduling.

II. PROPOSED ARCHITECTURE

Fig. 1 illustrates the proposed architecture, called *Time-domain Wavelength Interleaved Networking (TWIN)*, which consists of *aggregation devices (ADs)* at the edge of the network and *wavelength-selective cross-connects (WSXCs)* in the core. The AD provides service interfaces to different types of clients. The main functions of the AD are to aggregate incoming client protocol data units at the source/ingress, encapsulate them into *bursts*, and transmit the bursts to their respective destinations via a fast tunable laser. The fast tunable laser rapidly switches its wavelength depending on the destination of the burst¹. Specifically, a burst intended for destination j will be transmitted with wavelength λ_j at the source. Transmissions between two ADs within the network are optically transparent and bufferless. At destination, the AD demodulates the received optical signal, decapsulates each burst into respective client protocol data units, and delivers them to the appropriate clients. Typical traffic management functions (e.g., policing, shaping, queue management, and queue scheduling), adaptation functions (e.g., circuit emulation) and lookup tables (e.g., for VPN) can also be provided in the AD.

The network core constructed from WSXCs is optically transparent. The main function of the WSXC is to passively forward each optical burst arriving on its incoming fiber to its appropriate outgoing fiber based purely on the wavelength of the burst. No buffering or burst processing is provided at the WSXC. The burst forwarding process is said to be *self-routing* since no lookup table is involved in determining the next hop, thereby

¹A transmitter with a multifrequency laser can switch wavelengths in subnanoseconds [2].

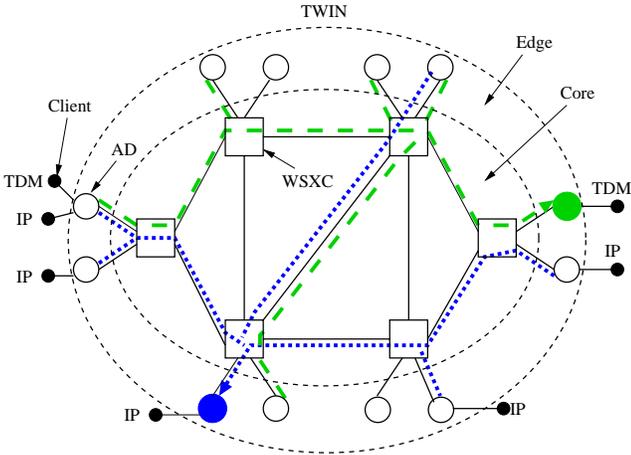


Fig. 1. The proposed architecture comprising access devices (ADs) at the edge and wavelength-selective cross-connects (WSXCs) in the core. Routing to a given destination follows a multipoint-to-point tree of a unique wavelength.

making the network core exceedingly simple. Another key advantage from an implementation standpoint is that wavelength-dependent forwarding can be realized without fast optical cross-connect reconfigurability, which currently is still a technological barrier. A micro-electromechanical system (MEMS)-based WSXC with four output ports and supporting 128 wavelength channels at 50 GHz spacing has been demonstrated in [3]. The route that bursts travel between each source and each destination in the network is determined through the configuration of each WSXC, which is provisioned at a relatively long time scale. One attractive approach is to select routes along a multipoint-to-point tree for each destination as shown in Fig. 1. Such a configuration ensures that if bursts do not collide at the destinations, they will not collide anywhere else in the network.

Since there is no buffering in the network core, each AD needs to perform media-access control for burst transmission. We propose to use *scheduling* to arbitrate burst transmissions at tunable lasers and ensure that *conflicts* (potential burst overlaps) do not occur in the network. The application of a multipoint-to-point tree considerably simplifies the scheduling problem as conflicts should be observed only at the transmitters and receivers, but not in the network core. One important objective of the scheduler is to maximize the achievable throughput of the network. In the context of a single node, *switch* scheduling in an input-queued crossbar switch has been investigated extensively in the past (e.g., [4], [5], [6], [7]). In the context of TWIN, we need to consider *network* scheduling that deals with the entire network consisting of multiple nodes. More importantly, network scheduling also has to explicitly take into account the propagation delays among various sources and destinations. These differences make the two problems significantly different.

A. Scheduling Cycles

Unlike switch scheduling where service configurations can be recomputed for each packet transmission, network scheduling with propagation delays needs to recompute service configurations at a longer time scale. The interval between two subsequent changes in service configurations is lower bounded by the

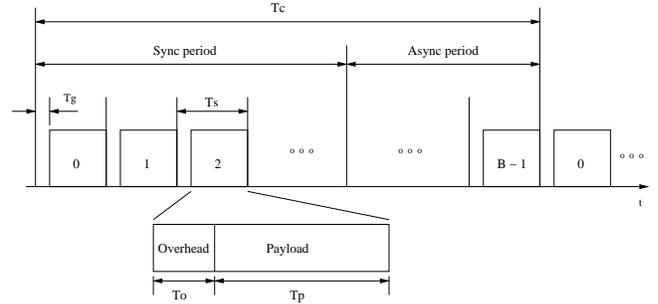


Fig. 2. Structure of the scheduling cycle.

propagation delay. It is desirable to assign the same rate to each connection within the interval. Thus, transmission of bursts can be facilitated through repetitive *scheduling cycles* of fixed duration T_c each, as shown in Fig. 2. A (scheduling) cycle consists of B time slots numbered from 0 to $B-1$, each of duration T_s . Each time slot can be used to carry one burst. Adjacent bursts are inter-spaced by a *guard time* of duration T_g to take into account time-of-day synchronization errors and other implementation factors. Each burst consists of an overhead of duration T_o and a payload of duration $T_p = T_s - T_g - T_o$. An important part of the overhead supplies a bit-synchronization preamble so that a receiver can perform frequency and phase synchronization to the transmitting bit stream. A burst-mode receiver capable of performing this synchronization within 50 ns has been successfully demonstrated [8]. Each cycle is divided into a *sync period* for transmission of synchronous (TDM) traffic and an *async period* for transmission of asynchronous (packet) traffic. The boundary between these two periods is flexible.

B. Timing for Scheduling

We assume that the scheduling cycle at each *destination* is aligned and synchronized to a common time-of-day clock, which can be derived from a GPS-based timing reference. A *schedule* for a given source-destination pair specifies one or more time slots that bursts from the source are to arrive at the destination in each cycle². Thus, if a source knows the schedule for a given destination and the propagation delay to the destination, the source can easily find the time to tune its laser to the destination. Suppose that time is expressed in units of time slot, taken to be unity. Consider the case where bursts from a particular source to a particular destination are to arrive at the destination in time slot t_a of each subsequent cycle. Assume that the propagation delay from the source to the destination is δ . Then the departure time of the next burst at the source is

$$t_d = \min_k \{t_a + kB - \delta \mid t_a + kB - \delta \geq t\} \quad (1)$$

where t is the present time and $k \in \mathbb{Z}^+$.

III. SCHEDULING

Scheduling is a key component of TWIN architecture. Scheduling could be implemented in a centralized or distributed fashion. In either case, scheduling could also be operated with

²A schedule is typically fixed for the entire duration of a connection with synchronous traffic, but may vary dynamically with asynchronous traffic.

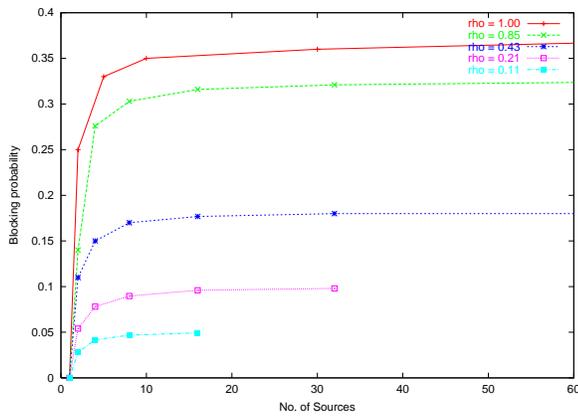


Fig. 3. Blocking probability versus number of sources as the batch size (number of transmissions per source) varies.

fixed or variable scheduling cycles. A centralized network scheduler for synchronous traffic that can achieve near-optimal performance has been proposed in [9]. A centralized scheduler with variable cycles has also been proposed in [10]. Here we focus on distributed scheduling for asynchronous traffic (e.g., packet traffic) with fixed cycles.

In its simplest form, distributed scheduling involves uncoordinated transmissions of bursts among sources. Each source, however, ensures that the bursts it transmits are not scheduled for transmission at the same time. The performance of this simple scheduler can be obtained by means of analysis or simulation (due to space constraint, the analysis cannot be provided in this paper). Figure 3 shows the performance of this scheduler in terms of burst blocking probability with a scheduling cycle of length $B = 150$. Consider a particular tagged destination. Each of the N sources is assumed to transmit d bursts in a cycle to the tagged destination. Some of these bursts will be blocked when they collide at the destination. The blocking probability is defined as the ratio of the number of bursts that are blocked to the total number of bursts transmitted (Nd). When Nd is fixed, we expect the number of bursts that are blocked increases as N increases, as shown in the figure. When $Nd = 150$ (or $\rho = 1.0$), observe that the blocking probability is about 35% as N becomes large. This also corresponds to the case when a source is completely uncoordinated. Note that the blocking probability drops to about 5% when $Nd = 16$ ($\rho = 0.11$).

We now describe an improvement to a distributed scheduler. The protocol relies on request and grant message exchanges to communicate schedules between a source and a destination, *learns* when collisions occur, and reassign time slots upon learning a collision. The basic idea of the protocol is illustrated in Fig. 4. In this example, the source initially requests burst transmission at the rate of three bursts per cycle. Upon receiving the request, the destination grants time slots 1, 6 and 9 for the source to use in the subsequent cycles. Upon receiving the grant, the source checks whether it can transmit the three bursts per cycle at the designated departure times on its tunable laser. Suppose the source can only transmit bursts on time slots 1 and 9, but not on 6 because of a conflict with another transmission. The destination learns about the conflict when an allocated time slot does not contain a burst. Upon detection of a conflict, the desti-

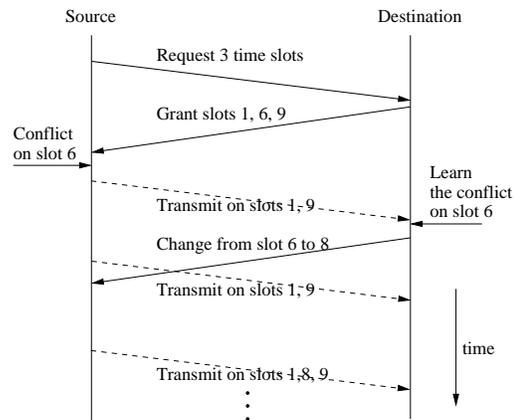


Fig. 4. The basic operation of the distributed protocol.

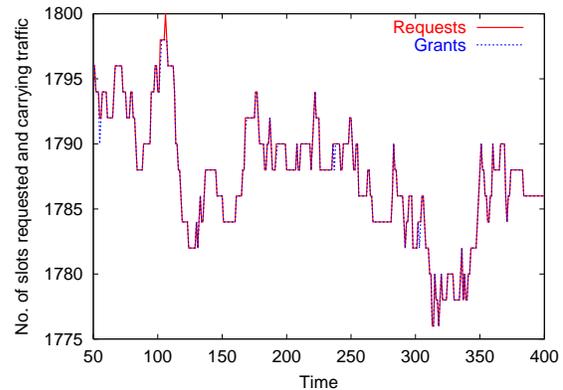


Fig. 5. Requests and grants as a function of time.

nation reassigns time slot 8 to the source. The source eventually transmits at its desired rate of three bursts per cycle.

Fig. 5 shows the dynamics of the distributed protocol with learning. Notice that the grants track the dynamics of the requests quite well, demonstrating the responsiveness of protocol with asynchronous traffic.

REFERENCES

- [1] L. Berger et al., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, Jan. 2003.
- [2] M. Kauer et al., "16-channel digitally tunable packet switching transmitter with sub-nanosecond switching time," in *Proc. ECOC*, paper 3.3.3, 2002.
- [3] D. M. Marom et al., "Wavelength-selective 1x4 switch for 128 WDM channels at 50 GHz spacing," in *Proc. OFC*, postdeadline paper FB7, Los Angeles, 2002.
- [4] A. Mekittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," in *Proc. IEEE INFOCOM'98*, Mar. 1998.
- [5] N. McKeown, "The iSLIP scheduling algorithm for input-queue switches," *IEEE Trans. on Networking*, vol. 7, pp. 188-201, Apr. 1999.
- [6] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proc. IEEE INFOCOM'00*, Aug. 2000.
- [7] C. Chang, W. Chen and H. Huang, "Birkhoff-von Neumann input buffered crossbar switches," in *Proc. IEEE INFOCOM'00*, Apr. 2000.
- [8] Y. Su et al., "Demonstration of a WDM-TDM metropolitan ring network prototype," Bell-Labs Technical Memorandum, Aug. 2002.
- [9] K. Ross, N. Bambos, K. Kumaran, I. Saniee, I. Widjaja, "Scheduling bursts in time-domain wavelength interleaved networks," Technical Report SU NETLAB-2002-12/1, Engineering Library, Stanford University, Stanford, CA 94305, Dec. 1992. Also to appear in *IEEE JSAC*.
- [10] K. Ross, N. Bambos, K. Kumaran, I. Saniee, I. Widjaja, "Dynamic scheduling of optical data bursts in time-domain wavelength interleaved networks," in *Hot Interconnects*, August, 2003.