An Addressing Independent Networking Structure Favorable for All-Optical Packet Switching

Shengming Jiang
School of Electronic & Information Engineering
South China University of Technology
shmjiang@scut.edu.cn

ABSTRACT

All-optical packet switching (AOPS) technology is essential to fully utilize the tremendous bandwidth provided by advanced optical communication techniques through forwarding packets in optical domain for the next generation network. However, long packet headers and other complex operations such as table lookup and packet header re-writing still have to be processed electronically for lack of cost-effective optical processing techniques. This not only increases system complexity but also limits packet forwarding speed due to optical-electronic-optical conversion. Lots of work of improving optical processing techniques to realize AOPS is reported in the literature. Differently, this paper proposes a new networking structure to facilitate AOPS realization and support various existing networks through simplifying networking operations. This structure only requires an AOPS node to process a short packet header to forward packets across it with neither table lookup nor header re-writing. Furthermore, it moves high layer addressing issues from packet forwarding mechanisms of routers. Consequently, any changes in addressing schemes such as address space extension do not require changes in the AOPS nodes. It can also support both connection-oriented and connectionless services to carry various types of traffic such as ATM and IP traffic. This structure is mainly based on the hierarchical source routing approach. The analytical results show that average packet header sizes are still acceptable even for long paths consisting of many nodes each of which has a large number of output ports.

Categories and Subject Descriptors: C.3 [Computer Systems Organization]: Computer Communication Networks

General Terms: Design.

Keywords: Networking structure, all-optical packet switching (AOPS), hierarchical source routing and addressing transparency.

1. INTRODUCTION

All optical packet switching (AOPS) technology has been studied for many years to improve the utilization of the tremendous bandwidth provided by advanced optical communication techniques. With AOPS, the signal is kept in optical domain for full optical signal processing to avoid delay caused by optical-electronic-optical conversion. Compared with electronic packet switching, optical packet switching has an additional wavelength domain for contention resolution to forward packets across switching fabrics. However, it is impossible to assign different wavelengths to many packets each with different destination for packet switching simultaneously because of a limited number of wavelengths available for such use. Similar to electronic packet switching, in order to switch packet by packet, optical packet switching also needs to process the packet header to decide an outgoing path across a node for each incoming packet. Thus, one of the most important issues for AOPS is how to process packet headers all optically for packet-by-packet forwarding.

1.1 Packet header processing

Due to the limited optical computing and buffering capabilities available today, all-optical-header processing is realized only in very simple forms [1]. It is almost impossible to all-optically process long packet headers such as the IP address. There is some inspiring progress in optical processing techniques reported recently. For example, in 2001, Lenslet (http://www.lenslet.com) announced the world first commercial re-configurable optics-based signal processing engine core. The first miniature photonic chip developed by some Australian researchers was reported in 2004 (http://www.pr.mq.edu.au). However, it may still take some time to make these techniques to be cost-effective to all-optically process long packet headers for AOPS.

At the time-being, long packet headers are usually extracted from packets and converted into electronic signal to be processed electronically. A typical example is the well-known optical burst switching (OBS) [2, 3]. With OBS, an end-device first collects the packets toward the same destination to form a 'burst'. Then a control packet carrying the destination of the burst is sent earlier than the burst itself to setup a light path to the destination. As suggested by many schemes, the burst is sent out after an offset time without any acknowledgements of the light path setting-up. The burst is supposed to travel optically along the light path set by the control packet. Many issues may affect the performance of an OBS network such as effective throughput over multi-hop paths. These issues include burst aggregation algorithms, offset time sizes, burst loss due to failed light path setting and congestion [4].

Some networking structures such as ATM and MPLS [5] indeed have short packet headers, which however are still too long to be processed all-optically in a cost-effective way. Furthermore, their packet headers need re-writing for switching, which cannot be easily realized now [1]. The self-routing address approach was also adopted for AOPS such as [6], which uses an output port bitmap called node address. Each output port of a node has one bit in the port bitmap, which is set to 1 only if the associated path goes through this node via this port. A path is composed of such port bitmaps for

each node along the path. One advantage of this approach is it requires only one single-bit processing for AOPS but at the expense of its large packet overhead which increases rapidly with the number of output ports supported per node and the number of nodes that a path goes through.

As discussed in this paper, an alternative effort to realize AOPS is to design a simple networking structure that can make use of some current optical processing techniques. This is further inspired by some advances in optical processing techniques feasible to process short packet headers. For example, an all-optical header recognition method and a packet self-routing scheme for a 6-bit address at 100 Gbit/s were reported in [7]. An all-optical header processing technique able to distinguish a large number of header patterns was demonstrated in [8] (more can be found in [9]-[13]). Some relevant work was also reported in the literature, such as all-optical buffers that can provide variable and adjustable queueing delays [14, 15], handling packet contention of two simultaneously arriving optical packets [10], all-optically decreasing packet time-to-live [16] and separating a packet header from its payload [17, 10]. Although those techniques cannot be comparable with electronic ones in terms of processing capability, it can be perceived that some practical techniques for simple computing and buffering in the optical domain will be available to realize AOPS with a simple networking structure.

1.2 Addressing issues

The popular IP network faces the address starvation problem for its further growth, and more arguments about IP can be found in [18]. This is because the IP address field was fixed while the Internet size has been going beyond the expectation of its original designers. Since the IP address combines the identity of an endpoint with the routing information corresponding to this point, expanding the IP address space requires implemental changes in IP routers to support the new addresses. For example, IPv6 [19] was proposed to provide a larger address space than IPv4. However, fully deploying IPv6 is costly and may take a long time since the networking units in the current Internet (e.g., hosts, routers and domain name servers) are all based on IPv4. Furthermore, it has not been proved that the IPv6 address space will be still sufficient in the future. This is simply because (i) it is almost impossible to accurately predict the Internet size in the future and (ii) whether every IP address can be used effectively depends on address allocation with respect to the user distribution. However, the fixed and hierarchical IP address albeit favorable for routing makes it difficult to allocate the addresses according to the user distribution. Hence, it is possible that the IPv6 network still needs to be changed again in case of its address space exhaustion.

To avoid costly network upgrading caused by addressing issues, one can separate routing information from the logical identity of nodes (refer to [20] for more discussion on addressing schemes) with the connection-oriented (CO) and source routing (SR) approaches [21]-[24]. With CO such as ATM, a connection is established according to the destination address through a connection setup process before any data transmission. However, CO cannot efficiently support connectionless services (CL) for pervasive data applications especially for short message transmission [25]. With SR, a route is expressed by a concatenation of the identities of all the networking units (e.g., IP address) that a packet goes through from a source to its destination. One disadvantage of SR is its large packet overhead used to carry such kind of routing information especially over long paths. However, as studied in [26], the current Internet is a small world. Its average length at the domain level is less than 4 while that at the router level is shorter than 10. Therefore, with a proper design of packet headers, the overall overhead for SR can be contained within an acceptable level for such a 'small Internet'. Furthermore, as discussed in [21], the source routing can simplify router implementation since it does not require a large routing table to cover the global address space but only a small one for its neighbors. Other advantages of SR include its stateless characteristics for scalability with neither per-flow information nor packet header re-writing.

1.3 A new networking structure

The above-mentioned superiorities of the source routing approach are exploited here to design a new networking structure based on the hierarchical routing approach [27]-[29], Two-Level Source Routing with Domain-by-Domain Routing (simply TLSR). It is designed to achieve the following objectives: (i) favorable for AOPS realization; (ii) supporting both CL and CO services to carry traffic from various types of networks such as IP and ATM; and (iii) supporting various addressing schemes by moving this issue to the edge of a network. The routing in TLSR consists of domain-level routing and port-level routing. The domain-level routing reduces packet overhead by folding long port-level routing information into shorter one at the domain-level. It also supports end-to-end QoS at the domain-level while simplifying QoS support in the AOPS node. The port-level routing is processed all-optically in each AOPS node. Thus, the packet header for the port-level routing should be designed as simple as possible in order to facilitate AOPS realization. How simple it should be depends on manufacture's optical processing techniques while its size can be determined by manufactures themselves for their devices.

To facilitate AOPS realization, TLSR avoids using any table (e.g., λ-slot interchange or packet forwarding tables) and re-writing packet headers since they require more complex computation. To this end, the output port for an incoming packet to exit an AOPS node is carried directly by the packet header for the port-level routing. In the literature, there are many fabric switching techniques handling how to automatically and efficiently forward a packet within a node such as the Manhattan [30] and wormhole routing [31] networks as well as the work reported in [32], which proposes all-optical swapping techniques using electronic computing and tables in core routers. These techniques can be used to design switches or routers for ATM and IP networks. Similarly, they may also be enhanced to support AOPS in the network architecture proposed here. For an illustration, this paper provides an example on how to realize AOPS with the proposed architecture by using the Banyan network in Section 3.2. However, a deep discussion on this issue is out of the scope of this paper. Furthermore, it is also possible to extend the above mentioned switching techniques for routing in wild area networks, which are different from the adopted source routing approach here by their definitions.

The networking structure proposed here shares some similarity with those reported in the literature such as Pip [33] in terms of using the source routing approach. That is, the routing information is arranged in a series of the "indexes" of

the nodes to be visited in the order of the visiting sequence, which is called Forwarding Table Index Field (FTIF) chains in Pip. The major difference is, in Pip, the index is the high layer logical identity and FTIF is used for forwarding table lookup. In TLSR, the index is the physical output port index (OPI) of the node to be visited, which can completely avoid table lookup in an AOPS node for packet forwarding, and OPI is very short as discussed in Section 3.3.

The remainder of this paper is organized as follows. The proposed structure TLSR is introduced in Section 2, and its major implementation issues are discussed in Section 3. Some relevant issues such as addressing, network connection and end-to-end QoS are discussed in Section 4. Finally, the paper is summarized in Section 5.

2. TWO-LEVEL SOURCE ROUTING WITH DOMAIN-BY-DOMAIN ROUTING

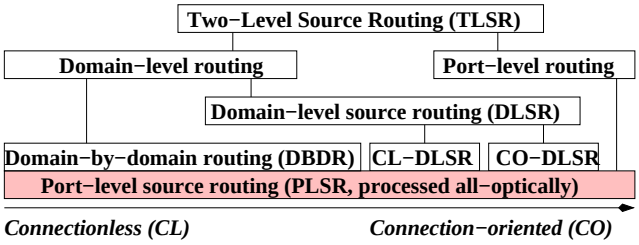

Figure 1: Hierarchical routing in TLSR

In TLSR, the routing is decomposed into the domain-level routing and port level routing as illustrated in Fig. 1. The networking unit of the domain-level routing is the domain while that of the port-level routing is the physical port of a node. The domain-level routing includes domain-by-domain routing (DBDR) and domain-level source routing (DLSR) while the port-level routing consists of only port-level source routing (PLSR). DLSR is further divided into connectionless DLSR (CL-DLSR) and connection-oriented DLSR (CO-DLSR) as discussed in the following sections. The route at the domain level is relatively stable since changes in domain linkages seldom happen. The domain-level routing does not need the details of a route across a domain in order to simplify routing management. However, the domain-level routing must be supported by the port-level routing since the routing across a domain is eventually performed by PLSR all-optically. Thus, with PLSR, the information used to route a packet across a node is explicitly carried in its packet header to avoid table lookup operation while no packet header re-writing is required for routing.

There are different definitions of taxonomies such as address and route in the literature [20, 34]. Here they are machine-friend and roughly defined below. The address indicates the identity of a host or a user in the network while a route is a road-map at the domain and/or port levels, which indicates how for a packet to travel between such addresses.

2.1 Domain-level source routing (DLSR)

DLSR is performed electronically at the ingress and/or egress of a domain. A DLSR route is expressed in a concatenation of either (i) the identities of all the domains that a packet will go through to reach its destination for CL-DLSR or (ii) the indexes of each intra-domain route along which a packet will travel across the domain for CO-DLSR. An intra-domain route is a series of the indexes of the output ports of each node that a packet is going to visit (refer to Section 2.3). Fig. 2(a) illustrates an example for CL-DLSR, in which an incoming packet is travelling from Domains 1 to 3 via 4. The numbers carried in the packet header indicate the identities of the next domains. Other information is ignored here for simplicity. After a packet passes the ingress of a domain, the identity of the corresponding next domain, which is located on the rightmost of the header, is stripped off from the header, and similar for CO-DLSR.

With CL-DLSR, upon a packet arriving in the first TLSR domain, its ingress needs to decide a concatenation of the next domain identities corresponding to the target domain of the packet, to which the destination node is attached. Usually links between domains are seldom changed. Thus, such concatenations corresponding to all target domains can be pre-setup. Each ingress needs to find one 'best' intra-domain route to cross the domain to reach the next domain indicated by the next domain's identity in the packet header (or the packet's destination if the current domain is the last one). This part is similar to the IP routing and can be realized with table lookup. However, the size of the next domain identity is much smaller than that of the IP address since the number of adjacent domains of a domain is usually very small. Therefore, this part can be performed very fast. After finding an intra-domain route, the ingress replaces this next domain filed with the particulars of this intra-domain route and passes the packet down to the PLSR layer.

With CO-DLSR, a domain connection is set up through a setup process, which tries to find a good intra-domain route in each domain between a pair of source and destination, and notifies the ingress of the corresponding domain with the selected intra-domain route. A domain connection is called dynamic domain connection if this process is invoked upon the arrival of such a request. Alternatively, some domain connections can be pre-setup with static intra-domain routes, in which changes in connectivity seldom happen. Such a domain connection is similar to the permanent ATM connection and called static domain connection here. The static domain connection can be set up either through the domain connection setting process or manually. Along such a setting process, resource reservation can also be made simultaneously to support QoS if necessary.

CL-DLSR is suitable for supporting CL services since CL usually does not require resource reservation especially for short message transmission. Alternatively, CL can also be supported by CO-DLSR with a static domain connection since a domain connection can be available upon a packet's arrival. There is hardly waiting delay incurred with a static domain connection whereas CL-DLSR needs some time to find an intra-domain route suitable for an arriving packet. However, CL-DLSR is helpful to balance traffic load since an intra-domain route can be selected according to the current traffic status in each node. For CO services, both the dynamic and static domain connections of CO-DLSR can be applied. For those requiring explicit resource reservation, a dynamic domain connection is the best choice.

Note that, a domain connection identity like VCI/VPI used in ATM can also be used here instead of the concatenation to reduce the overhead caused by DLSR. However, it requires storing per-flow state information which causes scalability problems in the core network. Furthermore, although DLSR is proposed to reduce the packet overhead of

the port-level routing and simplify end-to-end QoS support (see Section 4.3), this structure does fit the current Internet graph that can be logically divided into several autonomous systems, each of which has its own administrative control and resource management (pp. 432-433 in [5]).

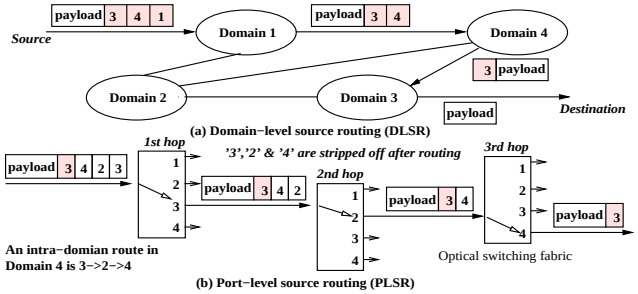

Figure 2: Overview of TLSR: DLSR and PLSR

2.2 Domain-by-domain routing (DBDR)

CL-DLSR needs a domain connection first to provide CL services across a domain by deciding an intra-domain route for a packet only upon its arrival. To efficiently support connectionless IP which is based on the node-by-node routing, DBDR is proposed. Different from CL-DLSR, with DBDR, the ingress of each domain decides the next domain according to the destination address only upon a packet's arrival at the domain without requiring a pre-setup domain connection. It is similar to the node-by-node routing but different in routing units, which are routers for the node-by-node routing while domains for DBDR. After deciding the next domain, the ingress tries to find an intra-domain route corresponding to this domain and forward the packet to it through PLSR. Since it is impossible to all-optically perform the node-by-node routing cost-effectively for long addresses as mentioned earlier, DBDR is a reasonable choice of supporting the node-by-node routing. Particularly for the IP network, the destination address carried in the IP header can be used here to determine the next domain. Furthermore, the Border Gateway Protocol (BGP) was extensively used by routers at the edge of the IP network to exchange routing information and traffic policies (pp. 459-461 in [5]). A protocol similar to BGP can also be developed for DBDR, which will be used by the domain ingress to exchange the domain-level routing information and traffic policies. In addition, DBDR can also make use of the routing information available for DLSR if any to find a concatenation of the next domains to reduce routing latency.

2.3 Port-level source routing (PLSR)

In an IP router, table lookup is required for every incoming packet to convert its IP address into the corresponding outgoing path across the router for routing. The similar operation is also needed in an ATM switch but with a much smaller table while re-writing the ATM header is also required for switching. It is very difficult to realize these operations all-optically as mentioned earlier. An output port source routing scheme was proposed in [23, 33] for electronic routers. Here, it is simplified for the all-optically processed PLSR. The networking unit of PLSR is the physical output port of a node. Accordingly, a PLSR-routed intra-domain route is expressed in a concatenation of the output port indexes (OPI) of all the nodes that a packet is going to visit. Thus, PLSR requires neither table lookup nor packet header re-writing operations to route a packet across a node, and only operation is to strip off, from the packet header, the OPI of the node currently visited by this packet. Therefore, such design is favorable for AOPS realization (see Section 3.1). Fig. 2(b) demonstrates how a packet travels along a PLSR intra-domain route going through three nodes in Domain 4, i.e., $3 \rightarrow 2 \rightarrow 4$, where 2, 3 and 4 indicates the OPIs of the corresponding nodes. Upon a packet arriving at a node, the optical switching fabric routes it directly to the corresponding output port indicated by the first OPI in the packet header. Once the packet reaches the corresponding output port, this OPI is stripped off.

Both DLSR and DBDR need the service of PLSR for routing packets optically across a domain while PLSR is used alone to route packets along an optical path. In this case, a packet always travels in an optical tunnel without going up for the electronically processed domain-level routing at the domain ingress. To this end, a packet needs to carry in its packet header the OPIs of all the intra-domain routes (rather than their indexes) adopted by each domain that the packet is going to visit in the order of its travel sequence. Such an optical route can provide a ultra-fast end-to-end transmission at the expensive of possible larger packet overheads for long routes consisting of many nodes.

3. IMPLEMENTATION ISSUES

This section discusses a preliminary packet header, a PLSR switching structure and packet header sizes for TLSR.

3.1 A preliminary packet header

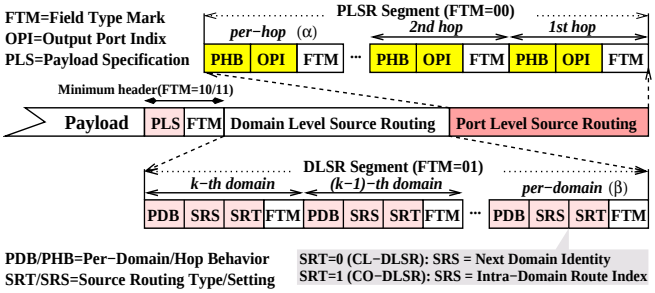

Figure 3: Formate of the TLSR packet header

As illustrated in Fig. 3, a variable packet header for TLSR consists of three parts, each of which starts with 'Field Type Mark' (FTM). FTM indicates whether its following field is for PLSR, DLSR, DBDR or other information. The first two segments of the packet carry the source routing information: one for PLSR and the other for DLSR. The third one is the minimum packet header consisting of FTM and Payload Specification (PLS). PLS specifies the type of the payload following itself as illustrated in Fig. 3 such as an IP packet or an ATM cell. The destination information carried by the payload is used by DBDR to decide the corresponding next domain. Such design tries to minimize per-packet header information and increases the versatility of TLSR since the PLS can indicate various protocols carried in the payload. Furthermore, the information carried in the payload (e.g., Total Length in IPv4) is re-used here to delimit a packet at the domain level. At the PLSR level, the per-node segment delimitation can be realized through guard time intervals

with a joint use of coding schemes as discussed in [17, 10].

As illustrated in Fig. 3, the per-domain DLSR segment beginning with FTM=01 is composed of Source Routing Type (SRT), Source Routing Set (SRS) and Per-Domain Behavior (PDB). This segment is repeated for every domain along a route. SRT indicates whether DLSR is for CL-DLSR (SRT=0) or CO-DLSR (SRT=1). SRS is set to the next domain identity for CL-DLSR while to the intra-domain route index for CO-DLSR. The next domain identity indicates the next domain of the current one that the packet is going to visit along a given route. An intra-domain route index is the identity of an intra-domain route used by the current domain to forward packets to the next domain along a given route.

The per-node PLSR segment starting with FTM=00 is composed of Output Port Index (OPI) and Per-Hop Behavior (PHB). OPI denotes the index of the output port through which the incoming packet is going to exit to reach its next hop. PHB indicates packet dropping preferences for congestion control. As discussed below, different formats of OPIs can co-exist for the same intra-domain route thanks to the nature of the source routing. How to use PHB and PDB to support end-to-end QoS is discussed in Section 4.3.

The DBDR segment beginning with FTM=10 is composed of only one PLS per packet. When a domain ingress receives such a packet, it first determines the format of the data carried in the payload pointed by PLS. Then, it learns the destination address from the payload according to the packet format. Given a destination address, the ingress can know whether the incoming packet needs to be forwarded to other domains to reach its destination. In this case, the ingress needs to find the next domain to the destination and an intra-domain route across this domain to reach the next domain. If the destination is located within the current domain, the ingress just forwards the packet to the destination accordingly. This forwarding maybe still goes through PLSR (if an intra-domain route is used to link the ingress and the destination node) or higher layer routing such as IP. FTM=11 can be used to indicate the end of an optical path.

3.2 Switching structure

The OPI explicitly indicates the outgoing path for a packet to cross a node so that no table lookup operation is required. The self-routing Banyan network [35] originally proposed for electronic switches can be used to realize an all-optical packet switch with such an OPI. As illustrated in Fig. 4, an optical signal arriving at a stage exits through Gate 0 to reach the next stage if its corresponding OPI bit is 0, and through Gate 1 if the bit is 1. For example, if the total number of output ports at a node is 8, the output port No. 4 can be expressed by OPI=100 in binary. A packet with OPI=100 from any input ports can be routed automatically to the output port No. 4 through a 3-stage optical fabric as as illustrated in Fig. 4. In this case, the bit 1 is first checked by the first stage, and then the first 0 bit by the second stage and so on as illustrated in this figure.

TLSR can provide a flexible OPI format to be defined by a node itself without need for standardization. This is because for PLSR, an OPI is only meaningful to a node that assigns it since this OPI will be used only by this node itself, which will strip off it after the packet reaches the corresponding output port. The OPIs and PHBs defined by other nodes will pass through this node cleanly at the bit level, and the same for amplifiers. Furthermore, all the labels (e.g., OPI,

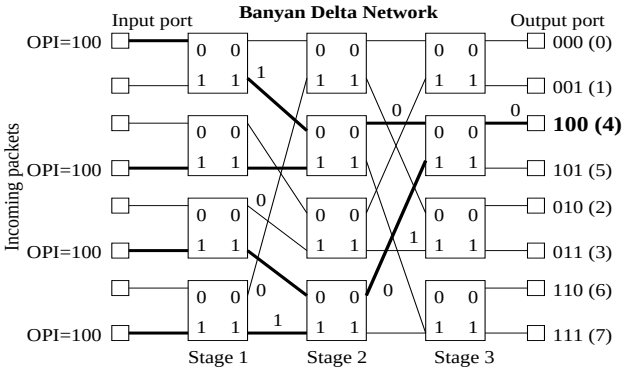

Figure 4: A PLSR realization (OPI=100)

PHB and PDB) are generated at domain-edge switches while the AOPS nodes within a domain do nothing. To this end, the domain-edge switch needs to learn the formats of all the OPIs defined by each particular AOPS node during the configuration process. This is feasible since the OPI format of a particular AOPS product does not change once it is defined. The OPI size affects the complexity and difficulty of the AOPS realization. Therefore, such flexible OPI can allow a manufacture to define its proprietary OPI according to its all-optical processing capability and product configuration. For example, the port bitmap proposed in [6] can also be adopted here for OPI, with which, the output port No. 0 is expressed by '1000...' while No. 4 by '0000100...'.

With TLSR, an AOPS node in a domain only processes PLSR according to OPI and PHB due to its poor optical processing capability as mentioned earlier. Other complex operations such as the label stacking and loop-back routing will be performed or assisted by domain-edge switches. For the label stacking, the encapsulation and stacked labels are carried in the TLSR payload. Therefore, this part only affects the domain-edge switch, which is similar to the MPLS switch discussed in the literature. A loop-back path can be formed by mapping the path into a concatenation of the OPIs of the switches along this path. Such a path can be determined by the domain-edge switch, and the same for the path-to-OPI mapping. Then an AOPS node will forward the related incoming packets to the corresponding output port indicated by the OPI carried in the packet header.

3.3 Packet header sizes and buffering

One major concern of the source routing based TLSR is its possible large packet header which increases with the number of nodes of a path. Given K domains and H nodes per domain that a path goes through, Appendix A analyzes the average and maximum sizes of the packet header, i.e., $A(K,H)$ and $M(K,H)$ given by (1)-(2), respectively. Table 1 lists $A(K,H)$ and $M(K,H)$ for (K,H) equal to $(4,3)$, $(10,25)$ and $(1,30)$, where γ, the PLS size (similar to the 4-bit 'version' in IP) is set to 8 bits and b is set to 32 delay bounds. Here $(4,3)$ is set according to the 'small Internet world' reported in [26], i.e., the average length of the Internet at the domain level is less than 4 while that at the router level is shorter than 10. So a route consists of 4 domains and $4\times3=12$ nodes. It can be found that $A(4,3)$ is equivalent to the ATM header's size. For both an all-optical path consisting of 30 nodes and the longest path comprising 250 nodes here, the TLSR header is shorter than or equivalent to the

Table 1: Maximum and average packet header sizes (in bits): $M(K,H)$ and $A(K,H)$

		$X=8$ ($\beta=12$)				$X=128$ ($\beta=16$)				$X=1024$ ($\beta=19$)				All-optical	
$(K,H)\rightarrow$		(4,3)		(10,25)		(4,3)		(10,25)		(4,3)		(10,25)		(1,30)	
x	(α)	$M()$	$A()$	$M()$	$A()$	$M()$	$A()$	$M()$	$A()$	$M()$	$A()$	$M()$	$A()$	$M()$	$A()$
16	(7)	67	32	293	103	79	32	329	104	88	34	356	104	220	119
256	(11)	79	38	393	155	91	40	429	156	100	42	456	156	340	181
4096	(15)	91	46	493	207	103	48	529	208	112	50	556	208	460	243
8192	(16)	94	48	518	220	106	50	554	221	115	52	581	221	490	258
Headers		MPLS=32				ATM=40				IPv4=192				IPv6=1280	

x: number of output ports per node, X: number of neighboring domains or intra-domain routes per domain.

IPv4 header, and much shorter than the IPv6 header on average, as indicated by $A(1,30)$ and $A(10,25)$. With the port bitmap based address structure-I proposed in [6], the overhead only for routing part is already $4096\times30=62880$ bits for (1,30) and 4096 ports per node (i.e., x) while the maximum header $M(1,30)=460$ bits only with TLSR. It can also be found that doubling the number of output ports per node (x) especially large x just results in a trivial increase in $A(K,H)$ and $M(K,H)$. Given x especially large ones, $A(K,H)$ changes slowly with X as shown in Table 1.

As indicated by $M(K,H)$ in Table 1, initially the entire packet header is large for long paths due to the nature of the source routing approach, which can be hardly buffered and processed all-optically. However, for the switching operation performed by an AOPS node, it only needs to check its α-long per-hop PLSR-segment in the packet header (see Fig. 3), which is always located at the beginning of each arriving packet. The remaining of the header is treated in the same way as for and together with the payload. That is, upon a packet signal arriving at an AOPS node, it first strips off its per-hop PLSR-segment from the packet (see Fig. 2) and buffers it if necessary for optical switching operation. The remaining of the packet (i.e., the remaining of the header plus the payload) may be buffered in an optical-loop buffer and then be passed through the node transparently once the path between the input port and the destination output port is established by the switching operation. As we know, the payload must be handled by all switching techniques. Here the difference is that the number of bits to be processed for packet forwarding with PLSR (α) is very small. The value of α depends on the number of output ports to be supported by a node. For example, to support 8192 ports per node along with a simple QoS capability, $\alpha=2$ bytes while only 7 bits for 16 ports per node as shown in Table 1. Processing such a PLSR-segment all-optically with neither table lookup nor header re-writing should be much easier than processing an IP address with table-lookup and than the labels of ATM and MPLS with both table-lookup and header re-writing.

4. DISCUSSIONS

This section discusses some important issues related to TLSR such as addressing, network connections and end-to-end QoS as well as domain-level congestion control.

4.1 Addressing issues

A reasonable answer to the question "whether an address space is sufficient" is to remove its limit. This does not mean to provide an infinite address space at any time. Instead, the address size should be adjustable without requiring any modifications to networking units. To reduce packet overhead for addressing, the address should not be fixed at the same size for all the users (here a user may refer to a person, a host or even a process). Its size can be set according to the user population while short addresses should be provided for wireless users to avoid header compression at the air interface. To facilitate routing, a hierarchical addressing structure similar to the telephone number is an option. That is, an address should consist of the user identity and high-level routing information. The former identifies a user while the latter indicates where the user is located in the network, and can be used to determine the domain connection. The user identity may consist of two parts: fixed and optional. The former is assigned by some organizations and cannot be changed either by the user itself or by the other parts. The latter can be defined by the user itself as a private address or by a third part as a temporary identity. Such private address can facilitate private network deployment such as ad hoc and sensor networks, which often require a rapid setup. The temporary identity is useful for mobility support as a mobile user roams from one place to another.

As discussed earlier, the routing units of TLSR include output ports and domains, which are the basic units of most existing networks. Hence, the routing in a TLSR network is not affected by the format and the size of the addresses adopted by higher layers. The conceptual differences in various addressing schemes only present at the edge of a TLSR network while all the packets are routed in the same way within each domain. For example, an IP gateway can be implemented at the edge of the TLSR network as illustrated in Fig. 5, which is responsible to convert an IP address into an optical route across the domain. In this case, the gateway is similar to an IP router. In an IP network, a packet needs to go through each router along a path to its destination. With TLSR, after passing the gateway, all the packets travel along a ultra-fast optical route. Any change in an implemented addressing scheme only needs to change the corresponding gateway without modification to the AOPS nodes within the TLSR network. Similarly, adding a new addressing scheme only needs to implement a gateway accordingly. Therefore, TLSR provides a flexibility for addressing schemes and allows a co-existence of multiple addressing schemes without requiring changes in the core network. The latter is important since different address schemes already exist today such as the IP address and the IEEE MAC address as well as the telephone number.

4.2 Network connections

As illustrated in Fig. 5, for a connection between two TLSR networks, if the incoming traffic is of PLSR (i.e., FTM=00), it is passed directly to the PLSR layer. This

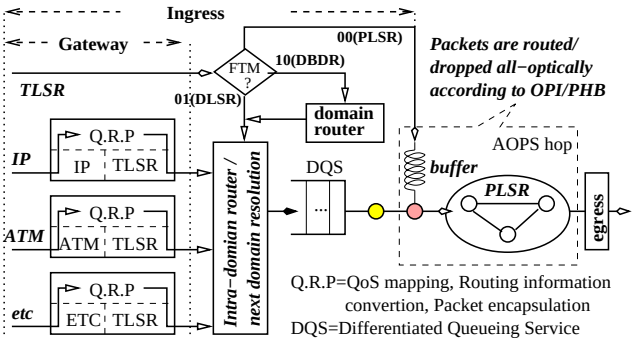

Figure 5: TLSR-based network connections

is the all-optical route discussed in Section 2.3. If the traffic is of DLSR (FTM=01), for CL-DLSR (SRT=0), the ingress needs to find an intra-domain route corresponding to the domain indicated by the first next domain identity carried in the packet header. For CO-DLSR (SRT=1), the intra-domain route is found through table lookup according to the first intra-domain route index carried in the header. For a DBDR packet beginning with FM=10, the domain router at the ingress needs to first determine the next domain based on the destination information carried in the payload, and then the corresponding intra-domain route across the domain. Similar to DBDR, 'FM=11' indicates the current domain is the final one and the destination can be determined with the information carried in the payload.

When a TLSR network connects with a non-TLSR network such as ATM, MPLS and IP, some conversion and parameter mapping between them are required, which are performed by the corresponding gateway located at the domain ingress. Such operations include QoS parameter mapping, routing information conversion and packet encapsulation as illustrated in Fig. 5.

QoS mapping converts the QoS parameters of non-TLSR networks into those of TLSR. If a TLSR network consists of multiple domains, it is the ingress of the first domain that decides Per-Domain-Behavior (PDH) for each domain along a route. PDH is determined according to the original QoS requirements such as end-to-end delay bounds and packet dropping rates. Then, the ingress of this domain decides PHB for each node along the selected intra-domain route, and the same for each sequential domains. More discussion on end-to-end QoS support can be found in Section 4.3.

Routing information conversion maps the routing information used by a non-TLSR network into the TLSR one, i.e., next domain identities, intra-domain route indexes and intra-domain routes. The first one can be used by DBDR to support CL services. Particularly for IP, a table can be maintained at the gateway to list IP addresses against next domain identities. As mentioned in Section 2.1, static CO-DLSR domain connections can also support CL services. To support CO services such as ATM, dynamic CO-DLSR domain connections can be used. In this case, the ATM connection setup process is followed by the TLSR domain connection setup process. A table listing ATM VCI/VPI against series of intra-domain route indexes is maintained at the gateway to map VCI/VPI into intra-domain routes.

For packet encapsulation, two approaches can be adopted: tunnelling and short-cut. With the first one, a packet from a non-TLSR network is encapsulated directly into TLSR's payload without changing the packet. Its major advantage is simplicity but at the expense of long packet overheads. This approach is more suitable for CL services since the original routing information may be still useful for routing if a packet will continue travelling in a non-TLSR network. With the second one, the original routing or switching information (e.g., IP addresses, ATM and MPLS lables) is removed from a packet and only its remaining is encapsulated into TLSR's payload. This approach is suitable for CO services since the removed information is often useless within an established channel after a connection is set up [36, 37].

4.3 End-to-end QoS

TLSR decomposes the effort to support end-to-end QoS into domain-level and node-level by jointly using classical QoS mechanisms. They include call-level call admission control (CAC) and packet-level scheduling and buffering algorithms [38]. The QoS support in the Internet evolutes from IntServ [39] to DiffServ [40], which suggests moving sophisticated QoS mechanisms from the core to the edge of a network in order to keep the core as simple and scalable as possible. This philosophy is also adopted for TLSR to further simplify QoS support at the node-level to facilitate AOPS realization. That is, CAC involves only the domain ingress and scheduling, and non-FIFO buffering schemes are implemented at the domain ingress too while only a simple or even zero buffer is required in an AOPS node. This is because the very limited all-optical processing capability is available now while the ingress and egress can perform complex operations with powerful electronic processing capability. Thus, the QoS support in an ingress is different from that in an AOPS node as discussed below.

At the domain level, each domain along a route is assigned with a portion of the task for the end-to-end QoS provisioning. This portion is expressed in terms of PDB, which is defined either through a call setup process or with a manual setting. Once PDB has been defined, each packet needs to carry this PDB in its header as illustrated in Fig. 3. All the packets are buffered and scheduled at the domain ingress according to PDB, which defines QoS requirements to be supported by this domain such as delay bounds and packet loss rates. Since a packet will travel along the all-optical tunnel which has a simple or zero buffer, bounding delay should be the major responsibility of a domain ingress. In this case, congestion control at the domain level (see Section 4.4) is important to reduce packet loss. Some research on PDB for DiffServ can be found in [41, 42].

At the node-level, a domain needs to distribute its QoS provisioning task to each node along a selected intra-domain route by properly defining PHB. However, for an AOPS node, PHB should be as simple as possible due to its limited optical processing capability. As mentioned above, delay and loss are mainly controlled by the domain ingress. A simple PHB can be defined for an AOPS node such as packet dropping preference for congestion control. A probabilistic dropping scheme [43] may be adopted for more sophisticate packet dropping. As illustrated in Fig. 3, PHB is carried along with OPI in the packet header. PHB should be standardized at the domain level since it is decided in the domain ingress and will be executed by nodes in this domain.

To support granular QoS, the differentiated queuing service (DQS) [44] can be adopted here. The basic idea of DQS is to allow each packet to carry its end-to-end QoS re-

quirements in its header so that each node can know these requirements and treat them accordingly. The required end-to-end delay is bounded by a node through queuing packets according to the delay bound assigned to this node. DQS suggests dropping those packets that are predicted 'unable to be guaranteed in terms of their delay bounds'. The sum of lost packets due to congestion and dropped packets due to delay overdue is guaranteed through CAC. Combining TLSR with DQS, we can project an end-to-end delay bound into delays to be bounded by each domain that a packet will go through. The delay bounded by a domain may be different from domain to domain, and is expressed explicitly as part of PDB carried in each packet header as illustrated in Fig. 3. The domain ingress queues all the incoming packets according to DQS [44] as illustrated in Fig. 5.

4.4 Domain-level congestion control

Similar to electronic switches, traffic congestion may also happen in all-optical packet switches. However, the impact of the congestion in an AOPS node on network performance is much more server than that in an electronic one due to much higher optical transmission rates. Such congestion can be released to some extent by using optical buffers. But due to the limited optical buffer capacity, the collision cannot be avoided completely. Similarly, due to the limited optical processing capability, a link-by-link retransmission is infeasible to retransmit lost packets in this case.

A possible domain-level solution to the problem caused by congestion consists of two parts. The first part is to use a TCP-like flow control scheme. That is, the domain-edge switches at both the ingress and egress of a domain can implement such a scheme to alleviate the congestion occurring in an AOPS node and retransmit lost packets (if any occurring in this domain) if necessary. The second part is to use some error control schemes to recover lost packets such as the forward error control (FEC) or multi-path transmission mechanisms. All these mechanisms implemented for a domain are transparent to the AOPS nodes inside the domain. This part needs a separate study.

5. SUMMARY

This paper discusses a new networking structure, Two-Level Source Routing with Domain-by-Domain Routing (simply TLSR). TLSR tries to simplify routing operations such that all-optical packet switching (AOPS) can be realized more easily and cost-effectively with limited all-optical processing techniques. It has the following characteristics.

-The port-level source routing (PLSR) does not require table-lookup and packet header re-writing to all-optically route a packet across an AOPS node. The information necessary for such routing is carried explicitly in the PLSR header without per-flow state information storage. The major components of the PLSR header can be defined by device manufactures without impact on connectivity with those from others. The PLSR header size mainly depends on the number of output ports of a node. The above two features make TLSR to be more easily realized all-optically compared with existing networking structures (e.g., IP and ATM).

-TLSR can support both connection-oriented (CO) and connectionless (CL) services. A connection between a pair of source and destination can be setup at both the domain and port levels, i.e., the domain connection and intra-domain route, respectively. For CO, a packet is processed electronically at the domain ingress to get an intra-domain route, and then travels along this intra-domain route optically to cross the domain with PLSR. An all-optical route between a pair of source and destination can also be setup so that a packet can avoid going up to the domain level. Both the domain-by-domain routing (DBDR) and connectionless domain level source routing (CL-DLSR) can be used to support CL services. The domain ingress can also select intra-domain routes dynamically according to the traffic situation.

-TLSR imposes no limitation on addressing schemes by moving the related issues to the edge of the TLSR domain. Any change in addressing schemes or address spaces requires no change in TLSR. Conceptual differences in various networking structures such as ATM, IP and MPLS only stand at the corresponding gateways located outside the TLSR network. Like DiffServ, TLSR also moves sophisticate QoS mechanisms to the network edge to make nodes as simple as possible to facilitate AOPS realization. Basically, the per-domain behavior (PDB) is defined to provide QoS at the domain level (e.g., delay bound and packet loss rate) while the per-node behavior (PBH) is defined for an AOPS node to fine adjust packet dropping in case of congestion.

-The packet header of TLSR is designed to minimize the information to be carried therein. The analytical results show that the average packet header size is not a major concern, which is equivalent to that of the ATM header for the current 'small Internet world' (refer to [26]). The average size for a route consisting of 10 domains each with an intra-domain route consisting of 25 nodes is still shorter than or equivalent to the IPv4 header and much shorter than the IPv6 header. The same is for an all-optical route consisting of 30 nodes (refer to Table 1).

Many issues on the practical realization of TLSR require more studies and some of them are listed below. (1) How can an AOPS node be realized to provide PLSR? For example, it seems that the Banyan network [35] can be used to realize PLSR. It is interesting to work more for its implementation. (2) To facilitate AOPS realization, PHB is simplified such that it only provides fine adjustment on packet dropping in case of congestion. Sophisticated QoS mechanisms are moved to the edge of the TLSR network. In this case, how to provide end-to-end QoS especially packet dropping rate needs more studies since a limited or even zero optical buffer is available in AOPS nodes. (3) More studies are also required for a more practical packet header that considers preamble and guard intervals between PLSR segments as well as between consecutive packets for the implementation.

APPENDIX

A. ANALYSIS OF PACKET HEADER SIZES

For simplicity, we assume an intra-domain route in each domain consists of the same number of nodes (H), the same size of the per-node PLSR segment (α) and the same size of the per-domain DLSR segment (β). Please refer to Fig. 3 for the packet format and definitions of the related parameters.

The packet header size depends on the number of domains that a route goes through (K). Here the average per-node packet header, $A(K, H)$, is calculated, which is the ratio of the sum of the lengths of a packet header presents at each node to the total number of nodes that the packet visits. For DLSR, the header size is $(K-1)\beta$ in the first domain since only the information for the next domains need to

be carried. Due to the header stripping-off operation, it becomes $(K-2)\beta$ in the second domain and zero in the last one. Thus, the length sum of the DLSR header for a route consisting of K domains is $K(K-1)\beta/2$. For PLSR, the size of an intra-domain route in the first node is $H\alpha$ while α in the last node of the same domain. Therefore, the length sum of the PLSR header for an intra-domain route is $H(H+1)\alpha/2$ per domain. To be conservative, here an intra-domain route is assumed still necessary in the last domain to route a packet. Then, the total PLSR header length for K domains is $KH(H+1)\alpha/2$. The minimum header with a length of $(2+\gamma)$ appears in every packet, where γ indicates the size of PLS and FTM is 2 bits long. Therefore, we have

$$\begin{aligned} A(K,H) &= 2+\gamma+\frac{1}{KH}[K(K-1)\beta/2+KH(H+1)\alpha/2] \\ &= 2+\gamma+0.5\alpha(1+H)+0.5(K-1)\beta/H. \quad (1) \end{aligned}$$

The maximum packet header is that of the packet present at the ingress of the first domain of a path since this packet needs to carry both DLSR and PLSR segments of all the next domains along this path. The DLSR segment is $(K-1)\beta$ bits long as mentioned above. Due to the $H\alpha$ bits for the PLSR segments of all the nodes in the first node and the $(2+\gamma)$ minimum header in every packet, the size of this maximum header, $M(K,H)$, is given by

$$M(K,H) = 2+\gamma+(K-1)\beta+H\alpha. \quad (2)$$

When $K=1$ and $H>0$, (1)-(2) refer to an all-optical route consisting of H AOPS nodes.

$A(K,H)$ increases lineally with K. From $\partial A(K,H)/\partial H = \alpha/2-(K-1)\beta/(2H^2)$, it can be found that $A(K,H)$ increases almost lineally with H too if K is small. For example, $A(1,H)$ for an all-optical channel is just a linear function of H with an increase factor of $\alpha/2$. For a large K, there is a turning point of H for the smallest $A(K,H)$, which is $H=\sqrt{(K-1)\beta/\alpha}$ given K by letting $\partial A(K,H)/\partial H=0$.

Now we discuss how to determine α and β by assuming that each node has the same number of output ports (x) while each domain has the same number of adjacent domains (y). The length of OPI is bounded by $\lceil \log_2^x \rceil$. PHB is set to 1 bit for packet dropping preference, similar to ATM. With FTM=2 bits, $\alpha=3+\lceil \log_2^x \rceil$. Similarly for β, one bit is enough for SRT. For packet dropping preference, it consists of two parts per domain, one for PDF and the other for delay bounds. Two bits are allocated for packet dropping preference to have more differentiations on packet dropping at the domain level. The delay bound depends on its expression format. Here a packet should carry the index rather than the exact value of a delay bound as proposed in [45] to reduce the overhead. That is, a domain assigns an index to each delay bound that it guarantees and this index is carried by PDB. Given b sets of delay bounds available per domain, the size of PDB is $2+\lceil \log_2^b \rceil$. For SRS, the size of the next domain identity is $\lceil \log_2^y \rceil$ for CL-DLSR. For CO-DLSR, the size of an intra-domain route index is $\lceil \log_2^z \rceil$, where z is the maximum number of intra-domain routes available per domain. With a 2-bit FTM, $\beta=4+\lceil \log_2^b \rceil+\lceil \log_2^X \rceil$, where $X=y$ for CL-DLSR and $X=z$ for CO-DLSR. Usually, z is larger than y since the number of neighbors of a domain is small.

B. REFERENCES

[1] K. Seppänen, "Optical time-division packet switch," [Available online] http://www.vtt.fi/tte/tte21/optical/td-packet-sw.pdf.

[2] J.S. Turner, "Terabit burst switching," *J. High Speed Net.*, vol. 8, pp. 3–16, 1999.

[3] Q.M. Qiao and M. Yeo, "Optical burst switching (OBS) - a new paradigm for an optical Internet," *J. High Speed Net.*, vol. 8, no. 1, pp. 69–84, Jan. 1999.

[4] T. Battestilli and H. Perros, "An introduction to optical burst switching for the next generation Internet," *IEEE Optical Commun.*, pp. S10–S15, Aug. 2003.

[5] A.S. Tanenbaum, *Computer Networks*, Pearson Education International, forth edition, 2003.

[6] X.C. Yuan, V.O.K. Li, C.Y. Li, and P.K.A Wai, "A novel self-routing address scheme for all-optical packet-switched networks with arbitrary topologies," *IEEE J. Lightwave Tech.*, vol. 21, no. 2, pp. 329–339, Feb. 2003.

[7] D. Cotter, J.K. Lucek, M. Shabeer, K. Smith, D.C. Rogers, D. Nesset, and P. Gunning, "Self-routing of 100 Gbit/s packets using 6 bit 'keyword' address recognition," *IEE Electronics Let.*, vol. 31, no. 25, pp. 2201–2202, Dec. 1995.

[8] N. Calabretta, H. de Waardt, G.D. Khoe, and H.J.S. Dorren, "Ultrafast asynchronous multioutput all-optical header processor," *IEEE Photonics Tech. Let.*, vol. 16, no. 4, pp. 1182–1184, Apr. 2004.

[9] H. Uenohara, T. Seki, and K. Kobyayashi, "Four-bit optical header processing and wavelength routing performance of optical packet switch with optical digital-to-analogue conversion-type header processor," *IEE Electronics Let.*, vol. 40, no. 9, Apr. 2004.

[10] H.J.S. Dorren, M.T. Hill, Y. Liu, N. Calabretta, A. Srivatsa, F.M. Huijskens, H. de Waardt, and G.D. Khoe, "Optical packet switching and buffering by using all-optical signal processing methods," *IEEE J. Lightwave Tech.*, vol. 21, no. 1, pp. 2–12, Jan. 2003.

[11] A.E. Willner, D. Gurkan, A.B. Sahin, J.E. McGeehan, and M.C. Hauer, "All-optical address recognition in next-generation optical networks," *IEEE Optical Commun.*, pp. S38–S44, May 2003.

[12] B.Y. Yu, R. Runser andP. Toliver, K.-L. Deng, D. Zhou, T. Chang, S.W. Seo, K.I. Kang, I. Glesk, and P.R. Prucnal, "Network demonstration of 100 Gbit/s optical packet switching with self-routing," *IEE Electronics Let.*, vol. 33, no. 16, pp. 1401–1403, Jul. 1997.

[13] R.A. Barry, V.W.S Chan, K.L. Hall, E.S. Kintzer, J.D. Moores, K.A. Rauschenbach, E.A. Swanson, L.E. Adams, C.R. Doerr, S.G. Finn, H.A. Haus, E.P. Ippen, W.S. Wong, and M. Haner, "All-Optical Network Consortium-ultrafast TDM networks," *IEEE J. Selected Areas Commun.*, , no. 5.

[14] T. Sakamoto, K. Noguchi, R. Sato, A. Okada, Y. Sakai, and M. Matsuoda, "Varaible optical delay circuit using wavelength converters," *IEE Electronics Let.*, vol. 37, no. 7, pp. 454–455, Mar. 2001.

[15] Y.-K. Yeo, J.J. Yu, and G.K. Chang, "A dynamically reconfigurable folded-path time delay buffer for optical packet switching," *IEEE Photonics Tech. Let.*, vol. 16, no. 11, pp. 2559–2561, Nov. 2004.

[16] J.E. McGeehan, S. Kumar, D. Gurkan, S.M.R.M. Nezam, A.E. Willner, K.R. Parameswaran, M.M. Fejer, J. Bannister, and J.D. Touch, "All-optical decrementing of a packet's time-to-live (TTL) field and subsequent dropping of a zero-TTL packet," *IEEE J. Lightwave Tech.*, vol. 21, no. 11, pp. 2746–2571, Nov. 2003.

[17] M. Nord, S. Bjornstad, and M. Nielsen, "Demonstration of optical packet switching scheme for header-payload separation and class-based forwarding," in *Proc. IEEE Optical Fiber Commun. Conf. (OFC)*, Los Angeles, California, USA, Feb. 2004, vol. 1, pp. 2689–2693.

[18] P. Molinero-Fernádenz, N. McKeown, and H. Zhang, "Is IP going to take over the world (of communications)?," in *Proc. ACM SIGCOMM WS. Hot Topics in Networks (HotNets)*, Princeton, New Jersey, USA, Oct. 2002.

[19] 3COM, "Understanding IP addressing: everything you ever wanted to know," White paper, 2001.

[20] P. Francis, *Addressing in Internetwork protocols*, Phd thesis, University College London, Sep. 1989.

[21] J.H. Saltzer, D.P. Reed, and D.D. Clark, "Source routing for campus-wide Internet transport," in *Local Networks for Computer Communications*, A. West and P. Janson, Eds., pp. 1–23. North-Holland Amsterdam, 1981.

[22] R.C. Dixon and D.A. Pitt, "II. Source routing bridges addressing, bridging, and source routing," *IEEE Network Mag.*, vol. 12, no. 1, pp. 25–32, Jan./Feb. 1988.

[23] D.R. Cheriton, "Sirpent^{TM}: a high-performance internetworking approach," in *Proc. ACM SIGCOMM*, Austin, Texas, USA, Sep. 1989, pp. 158–169.

[24] I. Cidon and I.S. Gopal, "Control mechanisms for high speed networks," in *Proc. IEEE ICC*, Atlanta GA, USA, Apr. 1990, vol. 2, pp. 0259–0263.

[25] S.M. Jiang, "Logical ring with ATM block transfer to support connectionless services in ATM," in *IEEE ATM Workshop*, Fairfax, VA, USA, May 1998, pp. 154–158.

[26] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Review of Modern Physics*, vol. 74, no. 1, pp. 47–98, Jan. 2002.

[27] J. McQuillan, "Adaptive routing algorithms for distributed computer networks," Tech. Rep. BBN Rep. 2831, Bolt Beranek and Newman Inc, Cambridge MA, USA, May 1974.

[28] J. Behrens and J.J. Garcia-Luna-Aceves, "Hierarchical routing using link vectors," in *Proc. IEEE INFOCOM*, San Francisco, USA, Mar. 1998, vol. 2, pp. 702–710.

[29] M. Montgomery and G. De Veciana, "Hierarchical source routing through clouds," in *Proc. IEEE INFOCOM*, San Francisco, USA, Mar. 1998, vol. 2, pp. 685–692.

[30] S.R. Madila and D. Zhou, "Routing in general junctions," *IEEE Tran. Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 11, pp. 1174–1184, Nov. 1989.

[31] L.M. Ni and P.K. McKinley, "A survey of wormhole routing techniques in direct networks," *IEEE Computer*, vol. 26, no. 2, pp. 62–76, Feb. 1993.

[32] D.J. Blumenthal, B.E. Olsson, G. Rossi, T.E. Dimmick, L. Rau, M. Masanovic, O. Lavrova, R. Doshi, O. Jerphagnon, J.E. Bowers, V. Kaman, L.A. Coldren, and J. Barton, "All-optical label swapping networks and technologies," *IEEE J. Lightwave Tech.*, vol. 18, no. 12, pp. 2058–2074, Dec. 2000.

[33] P. Francis, "A near-term architecture for deploying Pip," *IEEE Network Mag.*, vol. 7, no. 3, pp. 30–37, May 1993.

[34] J. Shoch, "Inter-network naming, addressing, and routing," in *Proc. IEEE Comp Soc. Int. Conf.*, Sep. 1978, pp. 72–79.

[35] S. Sibal and J. Zhang, "On a class of Banyan networks and tandem Banyan switching fabrics," *IEEE Trans. Commun.*, vol. 43, no. 7, pp. 2231–2240, Jul. 1995.

[36] V. Firoiu, J. Kurose, and D. Towsley, "Performance evaluation of ATM shortcut cnnections in overlaid IP/ATM," CMPSCI Technical Report TR 97-40, University of Massachusetts, Department of Computer Science, University of Massachusetts, Jul. 1997.

[37] S.M. Jiang, Q.L. Ding, and M. Jin, "Flexible encapsulation for IP over ATM with ATM shortcuts," in *Proc. IEEE Int. Conf. Networks (ICON)*, Singapore, Nov. 2000, pp. 238–242.

[38] R. Guérin and V. Peris, "Quality-of-service in packet networks: basic mechanisms and directions," *Computer Net.*, vol. 31, no. 3, pp. 169–189, Feb. 1999.

[39] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview," *IETF RFC 1633*, Jun. 1994.

[40] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated dervices," *IETF RFC 2475*, Dec. 1998.

[41] Y.M. Jiang, "Per-pomain packet scale rate guarantee for expedited forwarding," in *Proc. Int. WS Quality of Service (IWQoS)*, Berkeley, CA, USA, Jun. 2003, pp. 422–439.

[42] R. Bless, K. Nichols, and K. Wehrle, "A low effort per-domain behavior (PDB) for differenticated services," RFC 3662, [Available on line] http://www.faqs.org/rfcs/rfc3662.html, Dec. 2003.

[43] L.H. Yang, Y.M. Jiang, and S.M. Jiang, "A Probabilistic Preemptive Scheme for Providing Service Differentiation in OBS Networks," in *Proc. IEEE Globecom*, San Francisco CA, USA, Dec. 2003, vol. 5, pp. 2689–2693.

[44] S.M. Jiang, "Granular differentiated queueing services for QoS: strcuture and cost model," *ACM SIGCOMM Comp. Commun. Reivew (CCR)*, vol. 35, no. 2, pp. 13–22, Apr. 2005.

[45] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," in *Proc. ACM SIGCOMM*, Cambridge MA, USA, Sep. 1999.