This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings

Adaptive Re-Routing Over Circuits: An Architecture for an Optical Backbone Network

Jerry Chou and Bill Lin
University of California San Diego, La Jolla, CA 92093

Abstract— **As Internet traffic continues to grow unabated at an exponential rate, it is unclear whether the existing packet routing network architecture based on electronic routers will continue to scale at the necessary pace. On the other hand, optical fiber and switching elements have demonstrated an abundance of capacity that appears to be unmatched by electronic routers. Although a number of optical backbone architectures have been proposed (e.g., optical burst switching), they generally rely on frequent dynamic circuit reconfigurations and new signaling protocols for network-wide coordination. Recently, we proposed an alternative optical backbone architecture called COPLAR based on a paradigm of coarse optical circuit switching by default and adaptive re-routing over circuits when necessary [4]. This approach is based on the provisioning of long-duration quasi-static optical circuits between IE (Ingress-Egress) router pairs at the boundary of the network to carry the traffic by default. When a provisioned circuit is inadequate, we adaptively load-balance the excess traffic across circuits with spare capacity so that all traffic can be routed to their final destinations without the need to create new circuits on-the-fly. Our initial work was focused on the system architecture design and the provisioning of quasi-static circuits. In this paper, we focus on the problem of adaptive re-routing over circuits. Our evaluation using real traffic data on two real backbone networks (Abilene and GEANT) shows that our adaptive re-routing over circuits approach can effectively accommodate excess traffic even under heavy traffic loads.**

I. Introduction

For the past decade, Internet traffic has been doubling every year, and there is no indication that this rate of growth will slow down in the near future. While the packet switching approach used in the Internet backbone networks has thus far been able to keep up, it is unclear whether electronic routers that have been used at the core of backbone networks will continue to scale to match future traffic growth or optical link rates. On the other hand, optical fiber and switching elements have demonstrated an abundance of capacity that appears to be unmatched by electronic routers. The rate of increase in optical transport capacity has been keeping pace with traffic growth (with 100 Gb/s per wavelength in the next generation). Thus, one way of keeping pace with future traffic demands is to build an all-optical backbone network. However, packet switching requires the buffering of packets, of which optical switches are not capable today, and it is unclear if these functions can be practically realized in optics. On the other hand, circuit switching has a much simpler data transport, making it suited to optics and its vast capacity potential.

To harness the huge capacity of optical circuit switching in an evolutionary way that is compatible with packet switching at the edge of the network, a number of candidate optical network data transport architectures have been proposed (e.g. [12], [10], [9]). From a bird's-eye view, these architectures all share a similar conceptual starting point in which the core of the network is an all-optical circuit-switched cloud, as depicted in Fig. 1. The optical circuit-switched cloud is comprised of long-haul DWDM links that are interconnected by optical cross-connects (OXC). Traffic traverses the circuit-switched cloud through pre-established circuits (lightpaths) at optical speeds. Boundary routers at the edge of the cloud provide a compatible packet switching interface to the rest of the Internet. The different proposed optical network data transport architectures differ in how they adapt to changing traffic conditions and the corresponding requirements on the granularity of circuits and the frequency of changes to the circuit configurations. Some of the approaches such as optical burst switching (OBS) [12] and TCP switching [10] are based on frequent changes to circuit configurations. Although the frequency of dynamic circuit reconfigurations imposed by these approaches is well within the capabilities of available optical switching technologies, the coordination of such frequent network-wide reconfigurations is not easy. Moreover, new signaling mechanisms and (electronic) control planes are required to facilitate the coordination.

Alternatively, we recently proposed an optical network architecture called COPLAR that is based on a new paradigm of *coarse optical circuit switching by default* and *adaptive re-routing over circuits when necessary* [4]. This approach is based on the provisioning of long-duration quasi-static optical circuits between IE (Ingress-Egress) router pairs at the boundary of the network to carry the traffic by default. By taking into account the statistical daily traffic variations observed in past traffic measurements, we showed that carefully pre-computed coarse circuit configurations can indeed accommodate the actual traffic most of the time. When a provisioned circuit is insufficient, our solution uses an adaptive load-balancing approach to re-route the (hopefully small amounts of) excess traffic over circuits with spare capacity. This way, all traffic can be transported to their final destinations without the need to create new circuits on-the-fly, with most traffic going through a direct circuit without re-routing.

978-1-4244-6739-6/10/$26.00 ©2010 IEEE

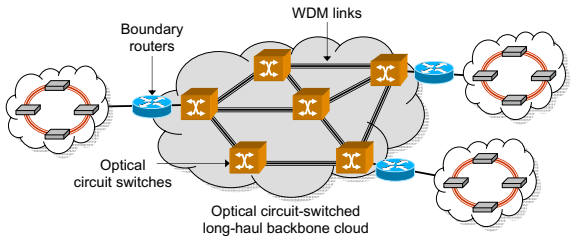


Fig. 1. Optical circuit-switched cloud with boundary routers.

Our initial work was focused on the architecture design [4] and the provisioning of quasi-static circuits [5]. In this paper, we focus on the problem of adaptive re-routing over circuits in the context of our COPLAR network architecture. Our problem differs from typical adaptive routing scenarios in packet switching networks in several important ways. First, in typical adaptive routing scenarios, the choice of routing paths is usually pre-defined and limited to a relatively small number of paths (e.g., all equal cost paths). However, as we will further explain in Section III-B, in the COPLAR routing architecture, a fully-connected mesh of circuits is typically provisioned to facilitate direct communications between any IE-pair, which leads to an exponential number of possible paths over circuits. Second, we need to ensure that the adaptive re-routing paths over circuits are *acyclic* to avoid having traffic stuck in loops.

To address these problems, we propose a novel method that can be used to derive and efficiently represent a set of acyclic routing paths over circuits. The proposed method aims to maximize the amount of traffic that can be re-routed over circuits with spare capacity by providing a high degree of path diversity. Using this method, we have incorporated a game-theoretic adaptive routing algorithm into our overall solution. We evaluated our proposed solution using real traffic data [11] on two real backbone networks, namely Abilene [1] and GEANT [3]. Our evaluation shows that our adaptive re-routing over circuits approach can effectively accommodate excess traffic even under heavy traffic loads.

The rest of this paper is organized as follows. Section II reviews related work. Section III briefly outlines the design of COPLAR. Section IV describes our approach to the selection and representation of adaptive re-routing paths over circuits. Section V describes our re-routing architecture. Finally, Section VI presents our evaluation.

II. Related Work

Several optical network data transport architectures have been proposed. Most approaches are based on frequent changes to circuit configurations. For example, in optical burst switching (OBS) [12], bursts of data are aggregated at the network edge by the boundary routers, and an out-of-band signaling process is used for establishing temporary circuits across the optical circuit-switched cloud for each burst. OBS adapts to changing traffic conditions by changing the circuit configurations on a frequent time-scale. Similarly, TCP switching [10] triggers the creation of circuit when it detects a new application (TCP) flow.

Another approach based on long-duration coarse circuits is to use a two-phase routing strategy [9]. Rather than configuring circuits to support a specific set of traffic matrices, two-phase routing optimizes circuits for the worst-case throughput under all possible traffic patterns permissible within the network's natural ingress-egress capacity constraints. However, optimizing for the worst-case leads to rather pessimistic performance.

Finally, motivated by the desire to adapt to changing traffic conditions, a number of online adaptive routing algorithms have been developed. TeXCP [8], MATE [6], and REPLEX [7] are representative examples. As already mentioned, our problem differs from the typical adaptive routing scenarios in packet switching networks in that the typical scenarios typically assume a limited pre-defined set of routing paths. These adaptive routing algorithms are complementary to our work in that we can extend them to work in our setting by incorporating mechanisms for the derivation and efficient representation of acyclic routing paths over circuits that provide ample path diversity. These adaptive routing algorithms also have to be extended to give preferences for direct circuit switching by default when sufficient circuit capacity is available.

III. COPLAR

This section briefly outlines COPLAR [4] by describing its core ideas of *coarse circuit switching* and *adaptive re-routing over circuits*.

A. Coarse Circuit Switching

For coarse circuit provisioning, the basic idea in COPLAR is to use historical traffic distributions as utility functions to model expected future traffic demands [4]. That is, we assume historical traffic demands during a particular time of day (e.g. 11am-noon on a weekday) are a good indicator of expected future traffic demands over the same time of day. The flows considered are at the level of IE-pairs where traffic between each pair of ingress and egress nodes of the network is considered as a commodity.

In particular, historical traffic demands can be explicitly captured by means of demand distribution functions $F = (f_i(x))$, with each $f_i(x)$ corresponding to the probability distribution of traffic demands for commodity C_i. For each $f_i(x)$, we have a corresponding cumulative distribution function (CDF) that describes the probability distribution of a random variable X that represents that actual traffic demand. Let x be the capacity of the provisioned circuit. Then the CDF of X is given by $\phi_i(x) = Pr[X \leq x]$, which corresponds to the probability that circuit capacity x is sufficient to satisfy the actual traffic demand X. To maximize the acceptance probability that the bandwidth allocations can satisfy the actual traffic demands for all commodities in a max-min fair manner, we can use these CDFs as utility functions. The problem can then be formulated and solved as a multi-path utility max-min fair bandwidth allocation problem, e.g. using the algorithm described in [5].

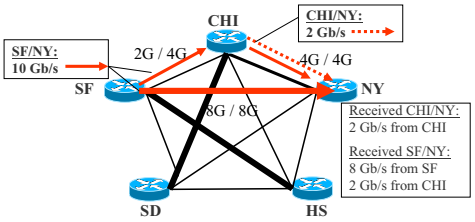


Fig. 2. When excess traffic occurs from SF to NY, we can re-route it using the residual circuit capacity of the path through Chicago.

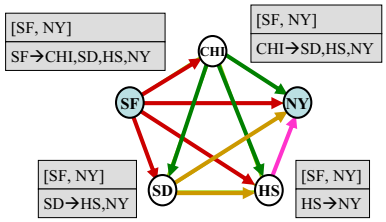


Fig. 3. To explore re-routing paths of each IE-pair, we construct routing tables based on an acyclic graph.

B. Adaptive Re-Routing Over Circuits

Although our circuit provisioning algorithm aims to maximize the probability that the actual traffic can be carried by direct circuits, traffic fluctuations or unexpected traffic changes can lead to inadequate capacities along direct circuits. Thus, the second part of COPLAR is to adaptively re-route the excess traffic over the provisioned circuits with sufficient spare capacity. Since our circuit provisioning algorithm described in the previous subsection is designed to create circuits between every IE-pairs, the logical network topology becomes a fully-connected mesh[1], as shown in Fig. 2.

Each link in Fig. 2 represents a logical circuit, which has the aggregated capacity from all its paths. For example, suppose the circuit capacity from SF (San Francisco) to NY (New York) is 8 Gb/s, and suppose the circuit capacities from SF to CHI (Chicago) and CHI to NY are both 4 Gb/s. Normally, we expect a circuit to have enough capacity for its direct traffic. For example, in Fig. 2, given 2 Gb/s of traffic from CHI to NY, all of its traffic can be directly sent through the network using the circuit from CHI to NY. However, suppose we have a 10 Gb/s burst of traffic from SF to NY, as shown in Fig. 2, then there would be 2 Gb/s of excess traffic because the circuit capacity from SF to NY is only 8 Gb/s. When this occurs, an adaptive re-routing mechanism is triggered to re-route the 2 Gb/s of excess traffic over alternative circuit routes, for example through CHI by the utilizing the residual circuit capacity available along SF-CHI and CHI-NY.

As can be seen from this example, with the help of adaptive re-routing, we can increase network throughput without the need to create new circuits on-the-fly. Although this adaptive re-routing does rely on electronic routing at intermediate nodes, it is only used as a secondary mechanism to handle excess traffic. The majority of traffic is still expected to be carried by the corresponding direct circuits. Therefore, simpler and much lower capacity routers can be employed. However, for this adaptive re-routing approach to be effective, two critical challenges must be overcome. First, we need to have an effective way to derive and efficiently represent the possible adaptive re-routing paths over circuits. Second, we need the actual hardware mechanisms to implement the adaptive re-routing. These challenges are described in the next two sections.

[1]i.e., if there is no traffic for a particular IE-pair, we can still think of it as having a direct circuit with zero capacity. Therefore, we simply use a fully-connected mesh to describe the logical network topology in the rest of the paper.

IV. Re-routing Path Exploration

As mentioned in Section III-B, the first step towards enabling adaptive re-routing over circuits is to derive the set of possible routing paths over circuits. Specifically, we need to explore available re-routing paths and efficiently represent them as route information. On the one hand, we would like to explore as many paths as possible to provide more path diversity for the re-routing algorithm and possibly achieve higher network throughput. On the other hand, more available paths usually also implies more route information has to be maintained in the network. In particularly, because our logic topology is a fully-connected mesh network, the maximum number of paths without routing loops between a single IE-pair is $O(N!)$. Thus, as network size increases, it becomes infeasible to maintain explicitly such exponential amounts of path information on routers using existing mechanisms such as MPLS.

To explore as many paths as possible without routing state explosion, we formulate the problem as a maximum acyclic graph problem. By finding a maximum acyclic graph between each IE-pair, it provides two important properties. First, all routing paths in a maximum acyclic graph can be implicitly represented by a set of routing tables at each node, and the number of entries in a routing table is linear to the network size instead of exponential, thus enabling us to avoid routing state explosion. Second, a maximum acyclic graph effectively maximizes the degree of path diversity while ensuring the avoidance of loops.

Since our circuit graph is a fully-connected mesh, a maximum acyclic graph of an IE-pair can be found in this special case by inducing any permutation order of nodes starting with its source and ending with its destination. Although all these acyclic graphs have the same number of paths, their maximum network throughput from source to destination is different because circuits do not have uniform capacity. For example, Fig. 3 shows one possible acyclic graph from SF to NY, but we can have another possible acyclic graph by changing the link between CHI and SD to the opposite direction. The two acyclic graphs may have different capacities if the circuit capacity from CHI to SD is not the same as the circuit capacity from SD to CHI. However, because our re-routing mechanism only utilizes the residual capacity at run-time, a circuit with larger capacity does not necessary provide more capacity

for re-routing. Therefore, how to select the best acyclic graph for re-routing is an interesting problem on its own. In our experiments, we formulated the problem as a *maxflow* problem for selecting the acyclic graph that achieves the maximum flow from the source to the destination so that there can be potentially more residual capacity left for re-routing. This *maxflow* problem is well-solved in the literature.

Once a maximum acyclic graph is selected, its routing paths can be represented by a set of routing tables at each node as follows. For every direct link (i, j) in the acyclic graph of IE-pair (s, t), we insert j as a next hop entry in the routing table of (s, t) on node i. This is depicted in Fig. 3 where the corresponding routing tables for the IE-pair SF-NY are shown at each router. For example, when a packet from SF to NY arrives at CHI, it can be forwarded to SD (San Diego), HS (Houston), or NY. By following these routing tables, packets can always be re-routed to their destinations in a loop-free manner as long as there is sufficient residual capacity along the circuit paths.

V. Re-routing Architecture

As depicted in Fig. 4, traffic for an IE-pair is sent via its direct circuit like traditional circuit switching when possible. When a packet arrives at an ingress point, it is first queued at the boundary router for traffic shaping, grooming, and buffering before the data is transferred by circuit frames/cells. For ease of explanation, we will refer to these queues as *circuit queues*, and we will refer to traffic that gets sent directly through circuits as *direct traffic*.

When the rate of traffic between an IE-pair is faster than its direct circuit capacity, the circuit queue would keep growing and eventually become *full*. Notice that the notion of "full" can be dictated by the queuing policy used[2]. Regardless of the queuing policy, when a packet cannot be inserted into its circuit queue, it is considered *excess* or *re-route* traffic, which is handled by the re-routing mechanism.

As shown in Fig. 5, when excess traffic occurs, it is adaptively re-routed to one of the circuits on its the next hop (corresponding to the acyclic graph construction). For example, the excess traffic from SF to NY can be re-routed through the SF-CHI circuit. Instead of inserting this re-route traffic into the circuit queue of the SF-CHI circuit, we queue this re-route traffic to the *standby queue* of the SF-CHI circuit. We give the packets in standby queue slower priority than those in the circuit queues. This prioritization means that direct traffic always takes precedence over re-route traffic. Alternatively said, the re-route traffic can only utilize residual capacity unused by direct traffic. Although the circuit and standby queues are shown as separate queues in Fig. 5, they can also be implemented as a single priority queue with two priority levels. WRED/RIO is one such implementation that is widely deployed in existing commercial routers [2], and it is also used in our experiments.

[2]e.g., random early drop policies can be used to provide some early indications of "full".

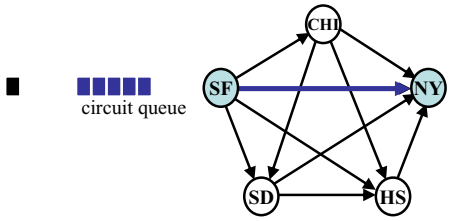


Fig. 4. Normally, a packet is queued at the network boundary, then transferred over its direct circuit through a network.

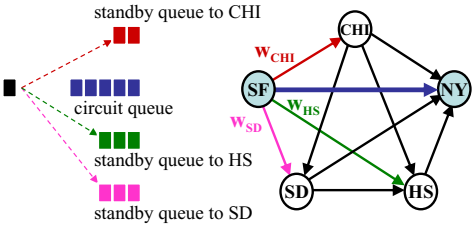


Fig. 5. When the circuit queue of a packet is full, the incoming packet becomes re-route traffic and is sent to the low priority standby queue of another outgoing circuit. The available next hop is decided by the routing table of re-routing paths, and an adaptive routing algorithm is applied to adjust the split ratios or weights among these paths.

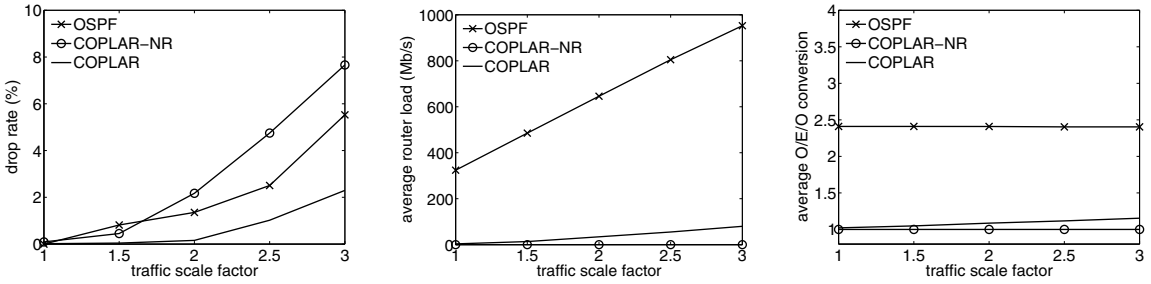
Finally, as shown in Fig. 5, for each IE-pair, our solution maintains a separate set of next-hop split-ratios at each intermediate node (e.g. W_{CHI}, W_{HS}, W_{SD}). These next-hop split-ratios are dynamically adjusted at run-time by an adaptive routing algorithm. The main objective of the adaptive routing algorithm is to balance re-route traffic through all re-routing paths so that network congestion can be alleviated. A number of existing adaptive routing algorithms can be adapted in our solution framework (e.g., [8], [6], [7]). Specifically, in our experiments, we adapted the REPLEX [7] algorithm to be the routing algorithm for several reasons. First, REPLEX is an effective adaptive routing algorithm that has been shown to achieve Wardrop equilibrium with fast convergence times. Second, REPLEX is applicable to our per-hop routing table structure, which does not require explicit path representations. REPLEX also has other strengths as mentioned in the related work section. Due to space limitations, we refer the reader to [7] for more details on how REPLEX works.

VI. Experiments

A. Experimental Setup

We evaluated our adaptive re-routing over circuits mechanism in COPLAR using two separate real backbone networks, namely Abilene [1] and GEANT [3]. Due to space limitations, we only present the results for the GEANT network in this paper. GEANT is a public network with 23 nodes and 74 links varied from 155 Mb/s to 10 Gb/s. In our evaluation setup, we used actual traffic measurements that have been collected on the GEANT network [11].

For coarse circuit configurations, we applied our circuit provisioning algorithm [5] to the GEANT network using the historical traffic measurements over the period between 01/01/2005 to 04/10/2005. To evaluate the performance of our re-routing method, we simulated the network traffic



(a) Drop rate comparison (b) Router load comparison (c) Average O/E/O comparison

Fig. 6. Network performance comparison among COPLAR, COPLAR-NR and OSPF when traffic is scaled by a factor 1 to 3, in steps size of 0.5.

on a separate date – specifically the traffic on 04/11/2005. To demonstrate the performance of our solution under a highly utilized network setting, we normalized the traffic by scaling up the offer loads so that at least one link became saturated under standard OSPF routing.

Finally, we implemented our solution using the NS2 simulator. As mentioned in Section V, we used the NS2 WRED/RIO priority queuing module to implement the circuit and standby queues, and we implemented REPLEX [7] as our adaptive routing algorithm for deciding upon the traffic split ratios among the re-routing circuit paths.

B. Experimental Results

Our evaluation compares three routing strategies: COPLAR, COPLAR-NR, and OSPF. Both COPLAR and COPLAR-NR are based on our circuit switching approach. COPLAR shows the results using our method of adaptive re-routing over circuits, whereas COPLAR-NR shows the results without adaptive re-routing – i.e., traffic is dropped once its circuit queue is full. OSPF is the commonly used packet routing method over the physical network topology.

Fig. 6 (a) shows the results of drop rate comparison when the network traffic is scaled by a factor from 1 to 3, with the factor increasing in 0.5 increments. At each scale factor, we simulated the traffic over 8 different time intervals evenly scattered across the date 04/11/2005, and we plotted the average drop rates in the figure. With increasing traffic, we can also observe increasing drop rates for all routing strategies. Among the three strategies, COPLAR-NR has the higher drop rates. This is not surprising since the capacity of circuits cannot be shared among IE-pairs. On the other hand, COPLAR achieves the lowest drop rates because our re-routing mechanism is capable of utilizing a high degree of path diversity made available by the circuits with residual capacity. At the scale factor of 3, the drop rates of COPLAR, COPLAR-NR, and OSPF are 2.30%, 7.66% and 5.53%, respectively. Thus, with adaptive re-routing over circuits, the drop rates were significantly reduced over COPLAR-NR and OSPF by almost 70% and 58%, respectively.

To demonstrate that re-routing only introduces limited overhead on network routers, Fig. 6 (b) shows the comparison of average router loads, which is the amount of traffic re-routed through intermediate routers. COPLAR-NR has absolutely zero router load because no traffic is re-routed. On the other hand, the load of OSPF increases rapidly because it relies on intermediate routers to forward packets along the entire path from ingress to egress. Finally, although the load of COPLAR also increases, it grows at a much slower rate because the majority of traffic is sent through direct circuits, and our adaptive routing algorithm prefers paths with lower delays and fewer circuit hops. As a result, at the factor of 3, our re-routing strategy only has an average router load of 80 Mb/s, which is less than 10% of the router load experienced by OSPF (953 Mb/s).

Finally, Fig. 6 compares the average number of O/E/O conversions per packet. COPLAR-NR has exactly one conversion because all traffic goes through direct circuits. OSPF has 2.5 average O/E/O conversions regardless of the traffic scale because the routing paths are static. For COPLAR, although the number of O/E/O conversions does increase with increasing traffic load because of the re-routing of traffic, the increase is extremely low, with an average of only 1.15 O/E/O conversions.

REFERENCES

[1] Abilene network. `http://abilene.internet2.edu`.

[2] Distributed weighted random early detection. `cisco.mn/en/US/docs/ios/11_1/feature/guide/WRED.pdf`.

[3] GEANT network. `http://www.geant.net/`.

[4] J. Chou and B. Lin. Coarse optical circuit switching by default, re-routing over circuit for adaptation. *IEEE OSA Journal of Optical Communications and Networking*, 18(1):33–50, January 2009.

[5] J. Chou and B. Lin. Optimal multi-path routing and bandwidth allocation under utility max-min fairness. In *IEEE IWQoS*, 2009.

[6] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS adaptive traffic engineering. In *IEEE INFOCOM*, pages 1300–1309, 2001.

[7] S. Fischer, N. Kammenhuber, and A. Feldmann. REPLEX: Dynamic traffic engineering based on wardrop routing policies. In *ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2006. ACM. *Information Processing Letters*, 1994.

[8] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *ACM SIGCOMM*, 2005.

[9] M. Kodialam, T. V. Lakshman, J. B. Orlin, and S. Sengupta. A versatile scheme for routing highly variable traffic in service overlays and IP backbones. In *IEEE INFOCOM*, 2006.

[10] P. Molinero-Fernandez and N. McKeown. TCP switching: exposing circuits to IP. In *Hot Interconnects*, pages 43–48, September 2001.

[11] TOTEM. GEANT traffic matrices. `totem.info.ucl.ac.be/dataset.html`.

[12] M. Yoo, C. Qiao, and S. Dixit. Optical burst switching for service differentiation in the next-generation optical internet. In *IEEE Communications*, pages 98–104, February 2001.