Advances in Oblivious Routing of Internet Traffic

M. Kodialam, T. V. Lakshman, and Sudipta Sengupta

Abstract Routing is a central topic in networking since it determines the connectivity between users. Recently, with the growing use of the Internet for a wide variety of bandwidth intensive applications, including peer-to-peer and on-demand/real-time multimedia, it has also become important that routing accounts for the quality-of-service needs of applications and users. A research problem of much current interest is traffic-oblivious routing for ensuring that the network provides the needed quality-of-service despite uncertain knowledge of the carried traffic. Oblivious routing involves using pre-determined paths to route between each ingress-egress node in the network (typically an Internet domain) that do not change with changing traffic patterns. By removing the need to detect changes in traffic in real-time or reconfigure the network in response to it, significant simplification in network management/operations and associated reduction in costs can be achieved. Moreover, oblivious routing has the potential to make the Internet much more robust and predictable in the face of rapidly varying and unpredictable traffic patterns. Theoretical advances in the area have shown that oblivious routing can provide these benefits without compromising capacity efficiency. We survey recent advances in oblivious routing with a view towards its application in (intra-domain) Internet routing.

1 Introduction

As the Internet continues to grow in size and complexity, it becomes increasingly difficult to predict future traffic patterns. Many emerging applications for the Inter-

M. Kodialam, T. V. Lakshman
Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ, USA
e-mail: {muralik, lakshman}@alcatel-lucent.com

Sudipta Sengupta
Microsoft Research, Redmond, WA, USA
e-mail: sudipta@microsoft.com

net are characterized by highly variable traffic behavior over time. We have already seen the emergence (or, migration) of many services over the public Internet carried over the ubiquitous Internet Protocol (IP), including VoIP (voice-over-IP), peer-to-peer file sharing, video streaming, and IPTV. The bandwidth requirements of these applications are not only high but show marked variations, both temporally and geographically, compared to traditional telephony voice traffic patterns. This has reduced the time-scales at which traffic changes dynamically, making it impossible to extrapolate past traffic patterns to the future.

To enable such applications to operate seamlessly, it is imperative that the Internet routing infrastructure be reliable and robust and guarantee network performance in the face of highly unpredictable and rapidly varying traffic patterns. Oblivious routing, due to its lack of dependence on accurate traffic knowledge, has hence become an important topic of research. Traffic-oblivious routing has the potential to make the Internet far more reliable and robust in the face of dynamically changing traffic patterns.

An oblivious routing scheme is determined by a set of (multi-)path routes (with traffic split ratios) for each source-destination pair in the network. Traffic is routed along those paths regardless of the (current) traffic matrix. An instance of oblivious routing is thus completely described by specifying how a unit flow is (splittably) routed between each source-destination pair in the network. Computing an oblivious routing scheme so as to minimize various performance metrics under a given traffic variation model has been the underlying theme of research on this topic.

We classify the work in the literature on oblivious routing into two broad categories based on the traffic variation model. In the *unconstrained* traffic variation model, the traffic matrix can vary arbitrarily and oblivious routing provides *relative* guarantees for routing a given traffic matrix with respect to the best routing for that matrix. In the *hose constrained* traffic variation model, the traffic matrix can vary subject to aggregate network ingress-egress constraints and oblivious routing can provide *absolute guarantees* for routing *all* such matrices.

For the hose model, oblivious routing schemes can be further classified depending on whether the routing from source to destination is along "direct" (possibly multi-hop) paths or through a set of intermediate nodes (also called two-phase routing). In the first phase of two-phase routing, incoming traffic is sent from the source to a set of intermediate nodes and then, in the second phase, from the intermediate nodes to the final destination. We point out the benefits of two-phase routing in supporting statically provisioned optical layer in IP-over-Optical networks and indirection in specialized service overlays. The origins of two-phase routing can be traced back to Valiant's randomized scheme for communication among parallel processors interconnected in a hypercube topology, also known as Valiant Load Balancing [36].

The paper is structured as follows. In Section 2, we discuss some aspects of the inherent difficulty in measuring traffic and reconfiguring the network in response to changes in it and bring out the need for oblivious routing. In Section 3, we introduce the traffic variation and performance models. In Section 4, we cover oblivious routing under the unconstrained traffic variation model. In Section 5, we cover obliv-

ious routing for the hose constrained traffic model. In Section 6, we cover two-phase (oblivious) routing for the hose constrained traffic model. We summarize in Section 7. Throughout the paper, we assume that the network is denoted by $G = (N, E)$ with node set N and (directed) edge (link) set E where each node in the network can be a source or destination of traffic. Let $|N| = n$ and $|E| = m$. The nodes in N are labeled $\{1, 2, \ldots, n\}$. We use T to represent a traffic matrix where t_{ij} is the traffic rate between nodes i and j.

2 The Need for Traffic Oblivious Routing

In an ideal network deployment scenario where complete traffic information is known and does not change over time, we can optimize the routing for that single traffic matrix – a large volume of research has addressed this problem. However, real-world traffic in Internet backbones is highly variable and largely unpredictable. In this section, we discuss some aspects of the inherent difficulty in measuring traffic and reconfiguring the network in response to changes in it. The most important innovation of oblivious routing schemes is the ability to handle traffic variability in a capacity efficient manner through static preconfiguration of the network and without requiring either (i) measurement of traffic in real-time or (ii) reconfiguration of the network in response to changes in it.

2.1 Difficulties in Measuring Traffic

Network traffic is not only hard to measure in real-time but even harder to predict based on past measurements. Direct measurement methods do not scale with network size as the number of entries in a traffic matrix is quadratic in the number of nodes. Moreover, such direct real-time monitoring methods lead to unacceptable degradation in router performance. In reality, only aggregate link traffic counts are available for traffic matrix estimation. SNMP (Simple Network Management Protocol) [8] provides this data via incoming and outgoing byte counts computed per link every 5 minutes. To estimate the traffic matrix from such link traffic measurements, the best techniques today give errors of 20% or more [28, 40, 37]. Moreover, many of these methods are not suitable for measuring traffic in real-time due to their computation intensive nature.

The emergence of new applications on the Internet, like P2P (peer-to-peer), VoIP (voice-over-IP), and video-on-demand has reduced the time-scales at which traffic changes dynamically, making it impossible to extrapolate past traffic patterns to the future. Currently, Internet Service Providers (ISPs) handle such unpredictability in network traffic by gross over-provisioning of capacity. This has led to ISP networks being under-utilized to levels below 30% [17].

2.2 Difficulties in Dynamic Network Reconfiguration

Even if it were possible to track changes in the traffic matrix in real-time, dynamic changes in routing in the network may be difficult or prohibitively expensive from a network management and operations perspective. In spite of the continuing research on network control plane and IP-Optical integration, network deployments are far away from utilizing the optical control plane to provide bandwidth provisioning in real-time to the IP layer. The unavailability of network control plane mechanisms for reconfiguring the network in response to and at time-scales of changing traffic further amplifies the necessity of the static preconfiguration property of a routing scheme in handling traffic variability.

3 Traffic Variation and Performance Models

There are two categories of traffic variation and associated performance models that have been considered in the literature in the context of oblivious routing. These models differ in the way they define the performance objective for evaluating the "quality" of a given oblivious routing scheme.

3.1 Unconstrained Traffic Variation Model

In this model, the traffic matrix to be routed by the network is arbitrary. Given (finite) link capacities in the network, there may not be any feasible routing scheme for a given traffic matrix. We define the optimal (non-oblivious or traffic-aware) routing for a given traffic matrix T to be the routing scheme that minimizes the maximum link utilization (or, equivalently, feasibly routes the matrix $\lambda \cdot T$ for the maximum possible value of the scalar λ). If the maximum link utilization achieved by the optimal scheme for routing T is greater than 1 (or, equivalently, $\lambda < 1$), then the traffic matrix T is infeasible for the network (under any routing scheme). However, we can still compare the performance of a given oblivious routing scheme to that of the optimal (non-oblivious) routing for a given traffic matrix, and use the worst performance gap (over all traffic matrices) as a measure of performance for oblivious routing. This *relative guarantee* is called the *oblivious performance ratio* and is described formally in Section 4. The objective is to pick an oblivious routing scheme with the best performance ratio.

3.2 Hose Constrained Traffic Variation Model

In the hose traffic model, the total amount of traffic that enters (leaves) each ingress (egress) node in the network is bounded by given values. This model was proposed by [12] and subsequently used by [10] as a method for specifying the bandwidth requirements of a Virtual Private Network (VPN). Note that the hose model naturally accommodates the network's ingress-egress capacity constraints.

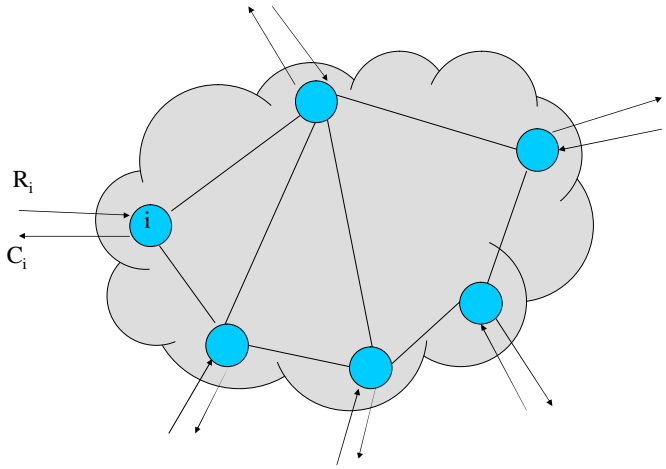

Fig. 1 Schematic of Network with Hose Constraints: Total ingress traffic at node i is upper bounded by R_i and total egress traffic at node i is upper bounded by C_i. Point-to-point traffic is not known.

Let the upper bounds on the total amount of traffic entering and leaving the network at node i be denoted by R_i and C_i respectively. This is illustrated in Figure 1. The point-to-point matrix for the traffic in the network is thus constrained by these ingress-egress capacity bounds. These constraints are the only known aspects of the traffic to be carried by the network, and knowing these is equivalent to knowing the row and column sum bounds on the traffic matrix. That is, any feasible traffic matrix $T = [t_{ij}]$ for the network must obey

$$\sum_{j \in N, j \neq i} t_{ij} \leq R_i, \quad \sum_{j \in N, j \neq i} t_{ji} \leq C_i \; \forall \; i \in N \tag{1}$$

For given R_i and C_i values, denote the set of all such matrices that are partially specified by their row and column sums by $\mathscr{T}(\mathbf{R}, \mathbf{C})$. We will use $\lambda \cdot \mathscr{T}(\mathbf{R}, \mathbf{C})$ to denote the set of all traffic matrices in $\mathscr{T}(\mathbf{R}, \mathbf{C})$ with their entries multiplied by λ. For the hose model, the throughput is defined as the maximum multiplier λ such that all matrices in $\lambda \cdot \mathscr{T}(\mathbf{R}, \mathbf{C})$ can be feasibly routed under given link capacities (by some routing scheme). The objective then is to choose an oblivious routing scheme

that maximizes the throughput. The problem of minimum cost network design has also been considered in which each link has associated costs per unit traffic. We will cover these problems in Sections 5 and 6.

4 Oblivious Routing under Unconstrained Traffic Model

The notion of oblivious routing was introduced by Räcke [29], in which demands are routed in an online manner and the objective is to minimize the maximum link utilization (or, congestion). Räcke proved the surprising result that it is possible to route obliviously in *any* undirected network and achieve a congestion that is within $\text{polylog}(n)$ times the optimal congestion for the specific set of demands (the latter being computed in an offline model). The proof, though constructive, relies on solving $\mathcal{NP}$-hard problems.

Earlier work had considered online routing so as to minimize congestion – in contrast to oblivious routing, this body of work uses current network state information (e.g., congestion on links) to route each demand. In [3], an online algorithm for this problem is designed that is $\log n$ competitive with respect to congestion.

Computing an oblivious routing so as to minimize the oblivious performance ratio has been the focus of majority of the work on this topic. We define this objective next, using the notion of throughput which is the reciprocal of maximum link utilization.

Consider a network with given link capacities. Let the *throughput* $\lambda(T)$ of a given matrix T be defined as the maximum multiplier λ such that the matrix $\lambda \cdot T$ is feasible (under some routing) for the given link capacities. This can be computed using a *maximum concurrent flow formulation* [32]. Note that the reciprocal of $\lambda(T)$ can be interpreted as the minimum (over all routing choices) of the maximum link utilization for routing matrix T.

Given an oblivious routing f, let $\lambda_f(T)$ denote the throughput of routing matrix T under f. Clearly, it follows that $\lambda(T) \geq \lambda_f(T)$. The performance of routing f on matrix T is worse than the best possible (traffic dependent) routing by a factor of $\lambda(T)/\lambda_f(T)$. The oblivious performance ratio of routing f is defined as the maximum value of this ratio over all traffic matrices.

$$\text{OBLIVIOUS-PERFORMANCE-RATIO}(f) = \max_T \frac{\lambda(T)}{\lambda_f(T)}$$

Following up from the above oblivious routing result by Räcke, polynomial time algorithms for oblivious routing (in undirected and directed networks) so as to minimize the oblivious performance ratio were developed in [4, 16]. In [4], the authors give a linear programming formulation with an exponential number of constraints and a polynomial time separation oracle for the constraints, thus leading to a polynomial time solution using the ellipsoid algorithm for linear programming [13]. Subsequently, a polynomial *size* linear programming formulation was developed in

[2] by taking the dual of the separation oracle and using it to replace the exponential set of constraints.

The oblivious performance ratio has been extended to handle restoration of link and node failures in [1]. The authors provide linear programming formulations for three different restoration models: (i) rerouting on the new network after failure, (ii) end-to-end restoration of paths affected by failure, and (iii) local (span) restoration around a failed link.

5 Oblivious Routing of Hose Constrained Traffic

Oblivious routing has also been considered in the context of the hose constrained traffic model described in Section 3.2. Since the hose traffic model bounds ingress-egress traffic at each node, the traffic on a network link under any oblivious routing choice is bounded. Hence, the relevant optimization problems in this context are minimum cost network design and minimizing the maximum link utilization (also referred to as maximum throughput network routing, as explained in Section 3.2).

Constant factor approximation algorithms for minimum cost network design for *single-path* oblivious routing are provided in [25, 15]. For this problem, the traffic usage on a link, for a given oblivious routing, is the maximum carried traffic on that link over all hose matrices. Link costs per unit traffic are given, and the objective is to minimize the total cost over all links.

In [11], the authors consider the problem of minimum cost (multi-path) oblivious routing of hose traffic under given link costs *and* link capacities. They give a linear programming formulation with an exponential number of constraints (and, a polynomial size separation oracle linear program) that is suitable for solving using the ellipsoid method [13]. The ellipsoid method is primarily a theoretical tool for proving polynomial-time solvability – its running time is not feasible for practical implementations. The authors also give a cutting-plane heuristic for solving the exponential size linear program and obtain reasonable running times for the experiments reported. However, this cutting-plane heuristic can have exponential running times in the worst case. In [23], the authors give a polynomial *size* linear program for maximum throughput oblivious routing of hose traffic under given link capacities. Their technique can be used to obtain a polynomial size linear program for the minimum cost version of the problem also, thus improving the result in [11].

Allocating link capacity for restoration under link failures has also been considered for oblivious routing of hose traffic. When the links used for routing form a tree topology, the problem of computing backup paths for each link and allocating capacity under a single link failure model is considered in [19], where a constant factor approximation algorithm is provided for minimizing total bandwidth reserved on backup links.

6 Two-Phase (Oblivious) Routing of Hose Constrained Traffic

Two-phase routing has been recently proposed [20, 38] as an oblivious routing scheme for handling arbitrary traffic variations subject to aggregate ingress-egress constraints (i.e., the hose constrained traffic model as described in Section 3). We begin with an overview of the scheme. As is indicative from the name, the routing scheme operates in two phases:

- **Phase 1:** A predetermined fraction α_j of the traffic entering the network at any node is distributed to every node j *independent of the final destination of the traffic*.
- **Phase 2:** As a result of the routing in Phase 1, each node receives traffic destined for different destinations that it routes to their respective destinations in this phase.

In contrast to two-phase routing, the oblivious routing approach in Section 5 can be viewed as routing along "direct" source-destination (possibly multi-hop) paths (ie., without the requirement to go through specific intermediate nodes). Hence, when comparing with two-phase routing, we will refer to the latter as "direct source-destination path routing". (Note that the use of the term "direct" in this context does not necessarily mean a single-hop path from source to destination.)

The traffic split ratios $\alpha_1, \alpha_2, \ldots, \alpha_n$ in Phase 1 of the scheme are such that $\sum_{i=1}^{n} \alpha_i = 1$. A simple method of implementing this routing scheme in the network is to form *fixed bandwidth paths between the nodes*. In order to differentiate between the paths carrying Phase 1 and Phase 2 traffic, we will refer to them as Phase 1 and Phase 2 paths respectively (Figure 2). The critical reason the two-phase routing strategy works is that the *bandwidth required for these tunnels depends on the ingress-egress capacities R_i, C_i and the traffic split ratios α_j but not on the (unknown) individual entries in the traffic matrix.* Depending on the underlying routing architecture, the Phase 1 and Phase 2 paths can be implemented as IP tunnels, optical layer circuits, or Label Switched Paths in Multi-Protocol Label Switching (MPLS) [31].

A subtle aspect of the scheme may not be apparent from its above description. Notwithstanding the two-phase nature of the scheme, some fraction of the traffic is actually routed to its destination in one phase, i.e., directly from source to destination (this may not be necessarily on a single-hop path). To see this, consider the traffic originating from node i and destined to node j. The fraction α_i of this traffic that should go to (intermediate) node i in Phase 1 does not appear on the network because it originates at node i. Hence, this traffic is routed directly to its destination in Phase 2. Similarly, a fraction α_j of the traffic that goes to (intermediate) node j in Phase 1 actually reaches its final destination after Phase 1. Hence, this traffic does not appear on the network in Phase 2. Thus, for each source-destination pair (i, j), a fraction $\alpha_i + \alpha_j$ of the traffic between them is directly routed to its destination using only one of the phases (either Phase 1 or Phase 2).

The problem of maximizing directly routed traffic in two-phase routing under either local (at a node) or global traffic information is considered in [27], for the

special case when the underlying topology is full-mesh, all ingress-egress capacities are equal, and all link capacities are equal.

We now derive the bandwidth requirement for the Phase 1 and Phase 2 paths. Consider a node i with maximum incoming traffic R_i. Node i sends $\alpha_j R_i$ amount of this traffic to node j during the first phase for each $j \in N$. Thus, the traffic demand from node i to node j as a result of Phase 1 routing is $\alpha_j R_i$. At the end of Phase 1, node i has received $\alpha_i R_k$ traffic from any other node k. Out of this, the traffic destined for node j is $\alpha_i t_{kj}$ since all traffic is initially split without regard to the final destination. The traffic that needs to be routed from node i to node j during Phase 2 is $\sum_{k \in N} \alpha_i t_{kj} \leq \alpha_i C_j$, where t_{kj} is the traffic rate from node k to node j. Thus, the traffic demand from node i to node j as a result of Phase 2 routing is $\alpha_i C_j$. This computation is shown in Figure 3.

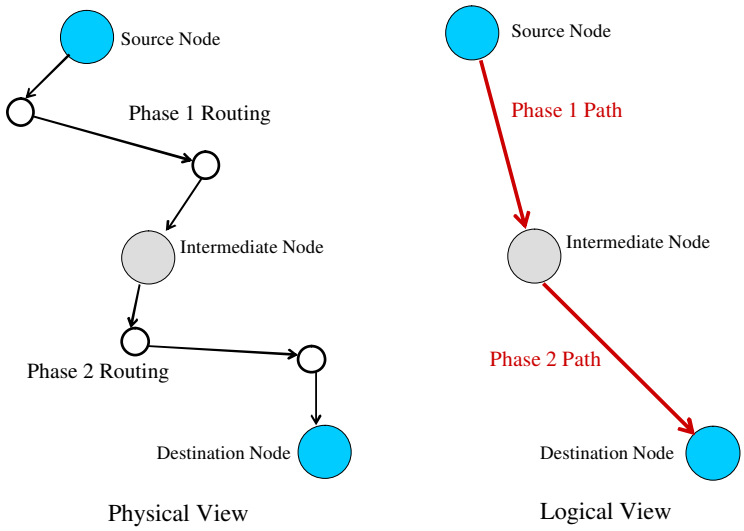

Fig. 2 A schematic figure of two-Phase Routing showing the physical and logical views. Note that the (logical) Phase 1 and Phase 2 paths can traverse multiple hops in the physical network.

Hence, the maximum demand from node i to node j as a result of routing in Phases 1 and 2 is $\alpha_j R_i + \alpha_i C_j$. Note that this does not depend on the matrix T. Two important properties of the scheme become clear from the above discussion. These are as follows:

Property 1 (Routing Oblivious to Traffic Variations): The routing of source-destination traffic is along fixed paths with predetermined traffic split ratios and *does not depend* on the current traffic matrix T.

Property 2 (Provisioned Capacity is Traffic Matrix Independent): The total demand from node i to node j as a result of routing in Phases 1 and 2 is $\alpha_j R_i + \alpha_i C_j$ and does not depend on the traffic matrix T but only on the aggregate ingress-egress capacities. The *bandwidth of the Phase 1 and Phase 2 paths* are *fixed*.

Property 2 implies that the scheme handles variability in traffic matrix T by effectively routing the fixed matrix $D = [d_{ij}] = [\alpha_j R_i + \alpha_i C_j]$ that depends only on aggregate ingress-egress capacities and the traffic split ratios $\alpha_1, \alpha_2, \ldots, \alpha_n$, and not on the specific matrix T. All that is required is that T be a feasible traffic matrix. This is what makes the routing scheme oblivious to changes in the traffic distribution.

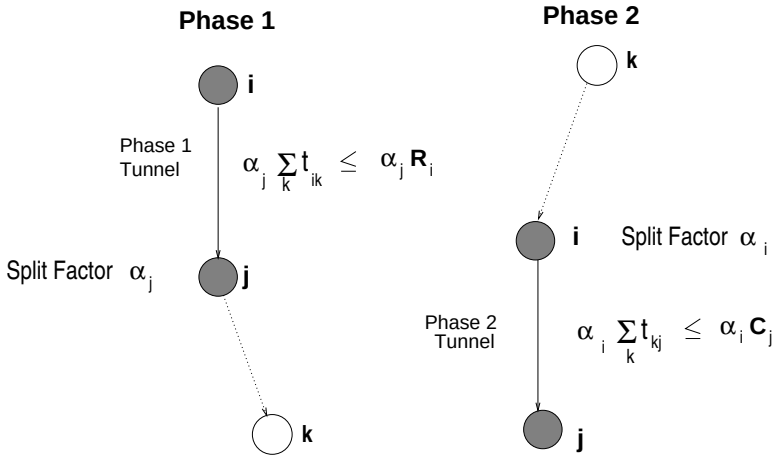

Fig. 3 Computing the bandwidth requirements for Phase 1 and Phase 2 paths between nodes i and j. In the bandwidth computation for Phase 1, node j is the intermediate node. In the bandwidth computation for Phase 2, node i is the intermediate node. Node k represents an arbitrary ingress/egress node.

We highlight below some additional properties of the two-phase routing scheme:

1. *Handling Indirection:* The routing decision at each source node in Phase 1 is independent of the final destination node of traffic. Hence, in addition to providing performance guarantees for variable traffic, two-phase routing is also ideally suited for providing bandwidth guaranteed services in architectures that decouple the sender from the receiver as in the *Internet Indirection Infrastructure* (i3) [35]. We discuss this further in Section 6.2.
2. *Multicast Traffic:* Two-phase routing can also naturally provide a *bandwidth guaranteed multicast* service. If the intermediate nodes can replicate packets for multicast, there is always sufficient bandwidth under two-phase routing to provide bandwidth guarantees for multicast traffic. The only requirement is that the ingress-egress capacity constraints of the hose model are satisfied by the total (unicast and multicast) traffic.

The origins of two-phase routing can be traced back to Valiant's randomized scheme for communication among parallel processors interconnected in a hypercube topology [36] where routing is through a *randomly and uniformly chosen intermediate node*. The two-phase routing scheme can be viewed as a deterministic scheme with possibly *unequal traffic split ratios* which can accommodate all traffic matrices within the network's natural ingress-egress capacity constraints. We will survey recent work on this scheme that has considered many new aspects arising from its potential application to routing Internet traffic in ISP backbone networks.

6.1 Addressing Some Aspects of Two-Phase Routing

We address some practical aspects of two-phase routing related to packet reordering and end-to-end delay and explain why they should not pose any hurdles in the deployment of the routing scheme in ISP networks.

Packet Reordering

In two-phase routing, the source node splits traffic to different intermediate nodes regardless of the final destination. Thus, packets belonging to the same end-to-end connection can arrive out of order at the destination node if the traffic is split within the same connection. The question arises whether this packet reordering is an issue that needs to be addressed.

The Internet standard for IP router requirements, RFC 1812, does not prohibit packet reordering in routers [5]. In fact, parallelism in router/OXC components and links causes packet reordering under normal operation and has been observed in the Internet [6, 18]. Packet reordering can affect the performance of TCP (Transmission Control Protocol) [9] and other traffic that relies on packet ordering. In its current version, TCP, which is used to carry most of the traffic in today's Internet, interprets packet reordering as a loss indicator. This triggers unnecessary retransmissions and TCP timeouts that cause a decrease in TCP throughput and increase in packet delay. Proposals have been made to make TCP more robust to packet reordering [7]. However, any new TCP feature requires protocol standardization and modification/upgrade of TCP software implementations running on hundreds of millions of client devices. Hence, it might be desirable to avoid packet reordering.

Packet reordering can be avoided in two-phase routing by splitting traffic at the application flow level at the source node (rather than at the packet level). An application level flow corresponds to a single end-to-end session of communication between two users on different machines and is commonly identified by a 5-tuple consisting of source IP address, destination IP address, source port number, destination port number, and protocol id [9]. In order to prevent congestion along the Phase 2 paths in two-phase routing, it is necessary to split the set of flows corresponding to each destination node among intermediate nodes in accordance with the traffic split ratios $\alpha_1, \alpha_2, \ldots, \alpha_n$.

The question then is whether two-phase routing with per-flow splitting can provide bandwidth guarantees to all traffic matrices within the network's natural ingress-egress capacity constraints. The answer, as we explain below, is in the affirmative. We are advocating this routing scheme for core networks where recent advances in DWDM (Dense Wavelength Division Multiplexing) transmission technologies [30] have resulted in link bandwidths of 10 Gbps and heading higher. Individual flows at best are in the Mbps range and hence small compared to link rates – tens of thousands of flows share a single 10 Gbps link. Moreover, TCP is not really good for gigabit flows – the throughput goes down as $1/(RTT * \sqrt{\text{random loss probability}})$ [9] and so very low random loss (due to bursty cross traffic) is needed to get gigabit throughputs for any reasonable RTT (round-trip time).

Increase in End-to-End Delay

Because of the two-phase nature of the routing scheme, packets might incur about twice the delay in end-to-end routing compared to direct source-destination path routing along shortest paths. However, for the portion of traffic that is routed directly to its destination (as explained in the description of the scheme), or when the intermediate node already lies on the shortest source-destination path, no additional delay is encountered, thus suggesting that the average delay may be less than a factor of two compared to that in direct source-destination path routing. In fact, experiments on actual ISP topologies collected for the Rocketfuel project [34] for maximum throughput two-phase routing indicate that the end-to-end hop count as a result of two-phase routing is about 1.4-1.6 times that of shortest path routing [33].

More importantly, this increase in delay may be tolerable for most applications. Consider an Internet backbone spanning the transcontinental US. A `ping` from MIT (US east coast) to UC Berkeley (US west coast) gives a round-trip time of about 90 msec. This round trip time includes two traversals of the long-haul network *and* two traversals each of Boston and Oakland metro access networks. Thus, the one-way end-to-end traversal time with two-phase routing deployed in the long-haul network can be expected to be much less than 90 msec. An end-to-end delay of up to 100 msec is acceptable for most applications.

Moreover, the bandwidth guarantees provided by two-phase routing under highly variable traffic reduce the delay variance (jitter). This reduction in jitter and the guarantee of predictable performance to unpredictable traffic provides a reasonable trade-off for the fixed increase in propagation delay. The effect of bursty traffic on jitter is mitigated by the source splitting of traffic to multiple intermediate nodes in two-phase routing.

Delay-sensitive traffic that cannot tolerate traversal of the core backbone twice can be routed along shortest paths in a hybrid architecture that accommodates both two-phase routing and direct source-destination path routing.

6.2 Benefits of Two-Phase Routing

In this section, we briefly discuss some properties of two-phase routing that differentiate it from direct source-destination path routing. We consider aspects of two different application scenarios to bring out the benefits of two-phase routing.

Static Optical Layer Provisioning in IP-over-Optical Networks

Core IP networks are often deployed by interconnecting routers over a switched optical backbone. When applied to such networks, direct source-destination path routing routes packets from source to destination along direct paths in the optical layer. Note that even though these paths are fixed a priori and do not depend on the traffic matrix, their *bandwidth requirements change* with variations in the traffic matrix. Thus, bandwidth needs to be deallocated from some paths and assigned to other paths as the traffic matrix changes. (Alternatively, paths between every source-destination pair can be provisioned a priori to handle the maximum traffic between them, but this leads to gross over-provisioning of capacity, since all source-destination pairs cannot simultaneously reach their peak traffic limit in the hose traffic model.) This necessitates (i) detection of changes in traffic patterns and (ii) *dynamic reconfiguration* of the provisioned optical layer circuits (i.e., change in bandwidth) in response to it. Both (i) and (ii) are difficult functionalities to deploy in current ISP networks, as discussed in Section 2. This amplifies the necessity of *static provisioning at the optical layer* in any scheme that handles traffic variability. Direct source-destination path routing does not meet this requirement.

To illustrate this point, consider the scenario in Figure 4 for direct source-destination routing in IP-over-Optical networks. Here, router A is connected to router C using 3 OC-48 connections and to Router D using 1 OC-12 connection, so as to meet the traffic demand from node A to nodes C and D of 7.5 Gbps and 600 Mbps respectively. Suppose that at a later time, traffic from A to C decreases to 5 Gbps, while traffic from A to D increases to 1200 Mbps. Then, the optical layer must be reconfigured so as to delete one OC-48 connection between A and C and creating a new OC-12 connection between A and D. As such, the *requirement of static provisioning at the optical layer is not met*.

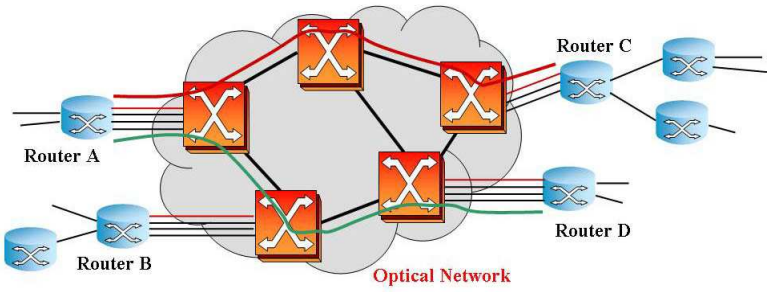

Fig. 4 Routing through direct optical layer circuits in IP-over-Optical networks.

Two-phase routing, as envisaged for IP-over-Optical networks, establishes the fixed bandwidth Phase 1 and Phase 2 paths at the optical layer. Thus, the *optical layer is statically provisioned* and does not need to be reconfigured in response to traffic changes. IP packets are routed end-to-end with *IP layer processing at a single intermediate node only*.

Indirection in Specialized Service Overlay Networks

The Internet Indirection Infrastructure (i3) was proposed in [35] to ease the deployment of services – like mobility, multicast and anycast – on the Internet. i3 provides a rendezvous-based communication abstraction through indirection – sources send packets to a logical identifier, and receivers express interest in packets sent to an identifier. The rendezvous points are provided by i3 servers that forward packets to all receivers that express interest in a particular identifier. The communication between senders and receivers is through these rendezvous points over an overlay network.

Two-phase routing can be used to provide bandwidth guarantees for variable traffic and support indirection in intra-ISP deployments of specialized service overlays like i3. (Note that we are not considering Internet-wide deployment here.) The intermediate nodes in the two-phase routing scheme are ideal candidates for locating i3 servers. Because we are considering a network whose topology is known, the two-phase routing scheme can be used to not only pick the i3 server locations (intermediate nodes) but also traffic engineer paths for routing with bandwidth guarantees between sender and receiver through i3 server nodes.

In service overlay models like i3, the *final destination of a packet is not known at the source* but only at the rendezvous nodes. Because the final destination of a packet needs to be known only at the intermediate nodes in two-phase routing, it is well-suited for specialized service overlays as envisaged above. In contrast, for direct source-destination path routing, the source needs to *know the destination of a packet* for routing it, thus rendering it unsuitable for such service overlay networks.

6.3 Determining Split Ratios and Path Routing

An instance of the scheme requires specification of the traffic split ratios $\alpha_1, \alpha_2, \ldots, \alpha_n$ and routing of the Phase 1 and Phase 2 paths. These can be picked and optimized for different performance objectives, given the network topology and ingress-egress constraints.

When link capacities are given, a common objective in the networking literature is to minimize the maximum link utilization, or equivalently, maximize the throughput. Linear programming based and combinatorial algorithms for computing the traffic split ratios and routing of paths so as to maximize network throughput are developed in [21]. The combinatorial algorithm uses a primal-dual technique based on a path-indexed linear programming formulation for the problem – it is simple to implement and involves a sequence of shortest path computations.

In [39], the authors consider the problem of minimizing the maximum *fanout* of any node when ingress-egress capacities are symmetric ($R_i = C_i$ for all i) and the underlying topology is full-mesh. The *fanout* of a node is defined as the ratio of its total outgoing link capacity to its ingress (egress) traffic capacity.

The problem of minimum cost network design (with the additional constraint of link capacities) can be formulated as a linear program and also admits solution by fast combinatorial algorithms [33].

6.4 Protecting Against Network Failures

We provide an overview of extensions to two-phase routing that can make the scheme resilient to network failures.

Router Node Failures in IP-over-Optical Networks

In an IP-over-Optical network where IP routers are interconnected over a switched optical backbone, the first and second phase paths are realized at the optical layer with router packet grooming at a single intermediate node only. IP routers are 200 times more unreliable than traditional carrier-grade switches and average 1219 minutes of down time per year [26]. Two-phase routing in IP-over-Optical networks can be made resilient against router node failures. (In the term "router node failure", node refers to an ISP PoP (Point-of-Presence), hence it includes the failure of all routers in a PoP.) When a router at a node f fails, any other node i cannot split any portion of its originating traffic to intermediate node f. Hence, it must redistribute the traffic split ratio α_f among other nodes $j \neq f$. In [22], the authors propose two different schemes for provisioning the optical layer to handle this redistribution of traffic – one that is failure node independent and static, and the other that is failure node dependent and dynamic.

Link Failures

Link failures can be caused by events like fiber cuts and malfunctioning of router/switch line cards. The two-phase routing scheme can be made resilient against link failures by protecting the first and second phase paths using *pre-provisioned* restoration mechanisms. By pre-provisioned, we mean that the backup paths are computed and "soft-reserved" a priori – the main action after failure is the rerouting of affected traffic on backup paths. The pre-provisioned nature of these mechanisms increases the reliability of the network by guaranteeing availability of backup resources after failure. It also allows fast restoration of traffic in an attempt to provide failure transparency to upper network layers. We discuss three such restoration mechanisms that have been considered in the literature for making two-phase routing resilient against link failures.

The first restoration mechanism [24] is local (or, link based) and consists of rerouting traffic around the failed link through pre-provisioned backup paths (link detours). The routing of traffic on portions of the primary path unaffected by the

failure remains unchanged. Backup paths protecting different links can share bandwidth on their links so as to guarantee complete recovery against any single link failure.

The other two mechanisms [24, 33] are end-to-end (or, path based) and different in the way backup bandwidth is shared across different link failure scenarios. In *K-route path restoration*, a connection consists of $K \geq 2$ link disjoint paths from source to destination. One of these paths is designated as the backup path and the others as primary paths. The backup path carries traffic when any one of the primary paths fail due to a link failure.

In *shared backup path restoration*, a primary path is protected by a link disjoint backup path. Different backup paths can share bandwidth on common links so long as their primary paths are link disjoint. Thus, backup bandwidth is shared to provide completely recovery against single link failures. Shared backup path restoration has been shown to have lower restoration capacity overhead compared to the other two mechanisms described above [14].

Linear programming based and combinatorial algorithms for maximum throughput two-phase routing under the respective protection schemes are presented in [22, 24, 33].

Complete Node Failures

We discussed how to make two-phase routing resilient to router node failures in IP-over-Optical networks. Failure of optical switches at a node in an IP-over-Optical network or of routers at a node in a pure IP router architecture (IP routers directly connected to WDM systems) leads to failure of the node for routing Phase 1 and Phase 2 paths also (in addition to loss of intermediate node functionality). The handling of such complete node failures in two-phase routing poses additional challenges. Failure of non-intermediate nodes lying on Phase 1 or Phase 2 paths can be restored by extending the mechanisms for protecting against link failures and using detours around nodes in local restoration or node-disjoint paths in path restoration. The failure of intermediate nodes can be handled through redistribution of traffic to other intermediate nodes. Because a complete node failure can lead to both of the above scenarios, a combination of the corresponding mechanisms can be used to protect against such failures.

6.5 Generalized Traffic Split Ratios

The traffic split ratios α_i can be generalized to depend on source and/or destination nodes of the traffic, as proposed in [20]. In this formulation, a fraction α_k^{ij} of the traffic that originates at node i whose destination is node j is routed to node k in Phase 1. While this does not meet the indirection requirement of specialized service overlays like i3, it can potentially increase the throughput performance of the two-phase routing scheme for other application scenarios like IP-over-Optical networks.

The problem of minimum cost network design or maximum throughput routing with generalized traffic split ratios is considered in [23]. For each problem, the authors first give a linear programming formulation with an exponential set of constraints and a corresponding separation oracle (linear program of polynomial size), thus leading to a polynomial time solution using the ellipsoid algorithm for linear programming [13]. Then, by using the dual of the separation oracle linear program to replace the corresponding set of exponential constraints, they obtain a polynomial *size* linear program for the problem.

An example to illustrate the improvement in throughput when the traffic split ratios are generalized as above is given in [23]. In fact, the same example shows that the gap between the throughput values of two-phase routing with intermediate node dependent traffic split ratios α_k and generalized traffic split ratios α_k^{ij} can be made arbitrarily large. However, the 2-optimality result for two-phase routing, which we discuss in Section 6.6, uses only intermediate node dependent traffic split ratios and assumes $R_i = C_i$ for all i. Hence, it follows that such pathological examples where the throughput improvement with generalized split ratios is arbitrarily large (or, even greater than 2) do not exist when ingress-egress capacities are symmetric.

Moreover, the pathological example in [23] exploits asymmetric link capacities and asymmetric ingress-egress capacities ($R_i \neq C_i$ for some nodes i). There is empirical evidence that suggests when ingress-egress and link capacities are symmetric, the *throughput of two-phase routing with α_i traffic split ratios equals that with generalized traffic split ratios and matches the throughput of direct source-destination routing along fixed paths*. This is indeed the case for the Rocketfuel topologies [34] evaluated in [23]. The above remains an open and unsettled question. In particular, it would be interesting to identify the assumptions under which this might be universally true (e.g., symmetric ingress-egress and link capacities).

6.6 Optimality Bound for Two-Phase Routing

Two-phase routing specifies ratios for splitting traffic among intermediate nodes and Phase 1 and Phase 2 paths for routing them. Thus, two-phase routing is one form of oblivious routing. However, as explained in Section 6.2, it has the desirable property of static provisioning that a general solution of oblivious routing (e.g., direct source-destination path routing) may not have. Moreover, when the traffic split ratios in two-phase routing depend on intermediate nodes only, the scheme does not require a packet's final destination to be known as the source, an indirection property that is required of specialized service overlays like i3.

A natural subject of investigation, then, is: *Do the desirable properties of two-phase routing come with any resource (throughput or cost) overhead compared to (i) direct source-destination path routing, and (ii) optimal scheme among the class of all schemes that are allowed to make the routing dynamically dependent on the traffic matrix?* This question has been addressed from two approaches.

First, using the polynomial size linear programming formulations developed in [23], the authors compare the throughput of two-phase routing with that of direct source-destination path routing on actual ISP topologies. Using upper bounds on the throughput of the optimal scheme, they compare the throughput of two-phase routing with that of the optimal scheme.

Second, the authors in [23] analyze the throughput (and cost) requirements of two-phase routing from a theoretical perspective and establish a 2-optimality bound. That is, the throughput of two-phase routing is at least $1/2$ that of the best possible scheme in which the *routing can be dependent on the traffic matrix*. It is worth emphasizing the generality of this result – it compares two-phase routing with the *most general class of schemes* for routing hose traffic. We briefly discuss this result.

Characterization of Optimal Scheme

Consider the class of all schemes for routing all matrices in $\mathscr{T}(\mathbf{R},\mathbf{C})$ where the routing can be made dependent on the traffic matrix. For any scheme $\mathscr{A}$, let $A(e,T)$ be the traffic on link e when matrix T is routed by $\mathscr{A}$. Then, the throughput λ_A of scheme $\mathscr{A}$ is given by

$$\lambda_A = \min_{e \in E} \frac{u_e}{\max_{T \in \mathscr{T}(\mathbf{R},\mathbf{C})} A(e,T)}$$

The optimal scheme is the one that achieves the maximum throughput λ_{OPT} among all schemes. This is given by

$$\lambda_{OPT} = \max_{\mathscr{A}} \lambda_A$$

For each $T \in \mathscr{T}(\mathbf{R},\mathbf{C})$, let $\lambda(T)$ be the maximum throughput achievable for routing the single matrix T. Then, the throughput of the optimal scheme can be expressed as

$$\lambda_{OPT} = \min_{T \in \mathscr{T}(\mathbf{R},\mathbf{C})} \lambda(T)$$

At first glance, the optimal scheme that maximizes throughput appears to be hard to specify because it can route each traffic matrix differently, of which there are infinitely many in $\mathscr{T}(\mathbf{R},\mathbf{C})$. However, because the link capacities are given in our throughput maximization model, (an) the optimal scheme can be characterized in a simple way from the proof of the lemma. Given a traffic matrix as input, route it in a manner that maximizes its throughput. Routing a single matrix so as to maximize its throughput is also known as the *maximum concurrent flow problem* [32] and is solvable in polynomial time. Clearly, the routing is dependent on the traffic matrix and can be different for different matrices.

The problem of computing λ_{OPT} can be shown to be $co\mathscr{NP}$-hard [33]. Computing the cost of the optimal scheme for the minimum cost network design version of the problem is also known to be $co\mathscr{NP}$-hard – the result is stated without proof in [15]. (An) The optimal scheme for minimum cost network design does not even

appear to have a simple characterization like that for maximum throughput network routing.

2-Optimality Result for Two-Phase Routing

The 2-optimal bound for two-phase routing that we now discuss establishes that two-phase routing provides a 2-approximation to the optimal scheme *for both maximum throughput network routing and minimum cost network design*. This is as an useful theoretical result, given the computational intractability of the optimal schemes for both problems.

Even though this theoretical result shows that the throughput of two-phase routing, in the *worst case*, can be as low as $1/2$ that of the optimal scheme (and, hence that of direct source-destination path routing), the experiments reported in [23] indicate that two-phase routing performs much better in practice – *the throughput of two-phase routing matches that of direct source-destination path routing and is within 6% of that of the optimal scheme on all evaluated topologies.*

For the 2-optimality result, it is assumed that ingress-egress capacities are symmetric, i.e., $R_i = C_i$ for all nodes i. This is not a restrictive assumption because network routers and switches have bidirectional ports (line cards), hence the ingress and egress capacities are equal.

Theorem 1. *Let $R_i = C_i$ for all nodes i, and $R = \sum_{i \in N} R_i$. Consider the throughput maximization problem under given link capacities. Then, the throughput of the optimal scheme is at most*

$$2\left(1 - \frac{1}{R}\min_{i \in N} R_i\right)$$

times that of two-phase routing.

The following result for minimum cost network design is also established in [23].

Theorem 2. *Let $R_i = C_i$ for all nodes i, and $R = \sum_{i \in N} R_i$. Consider the minimum cost network design problem under given link costs for unit traffic. Then, the cost of two-phase routing is at most*

$$2\left(1 - \frac{1}{R}\min_{i \in N} R_i\right)$$

times that of the optimal scheme.

7 Summary

We surveyed recent advances in oblivious routing for handling highly dynamic and changing traffic patterns on the Internet with bandwidth guarantees. If deployed, oblivious routing will allow service providers to operate their networks in a quasi-static manner where both intra-domain paths and the bandwidths allocated to these

paths is robust to extreme traffic variation. The ability to handle traffic variation without almost any routing adaptation will lead to more stable and robust Internet behavior. Theoretical advances in the area of oblivious routing have shown that it can provide these benefits without compromising capacity efficiency.

We classified the work in the literature on oblivious routing into two broad categories based on the traffic variation model – unconstrained traffic variation and hose traffic variation. For the hose model, we further classified oblivious routing schemes depending on whether the routing from source to destination is along "direct" (possibly multi-hop) paths or through a set of intermediate nodes (two-phase routing). Two-phase routing has the additional desirable properties of supporting (i) static optical layer provisioning in IP-over-Optical networks, and (ii) indirection in specialized service overlay models like i3.

References

1. D. Applegate, L. Breslau, and E. Cohen, "Coping with Network Failures: Routing Strategies for Optimal Demand Oblivious Restoration", *ACM SIGMETRICS 2004*, June 2004.
2. D. Applegate and E. Cohen, "Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs", *ACM SIGCOMM 2003*, August 2003.
3. J. Aspnes, Y. Azar, A. Fiat, S. A. Plotkin, and O. Waarts, "On-line routing of virtual circuits with applications to load balancing and machine scheduling", *Journal of the ACM*, 44(3):486504, 1997.
4. Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke, "Optimal Oblivious Routing in Polynomial Time", *Journal of Computer and System Sciences*, 69(3):383-394, 2004.
5. F. Baker, "Requirements for IP Version 4 Routers", RFC 1812, June 1995.
6. J. C. R. Bennett, C. Partridge, and N. Shectman, "Packet Reordering is Not Pathological Network Behavior", *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 789-798, December 1999.
7. E. Blanton and M. Allman, "On Making TCP More Robust to Packet Reordering", *ACM Computer Communication Review*, vol. 32, no. 1, pp. 20-30, January 2002.
8. J. Case, M. Fedor, M. Schoffstall, J. Davin, "Simple Network Management Protocol (SNMP)", RFC 1157, May 1990.
9. Douglas E. Comer, *Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture*, Prentice Hall, 4th edition, January 2000.
10. N. G. Duffield, P. Goyal, A. G. Greenberg, P. P. Mishra, K. K. Ramakrishnan, J. E. van der Merwe, "A flexible model for resource management in virtual private network", *ACM SIGCOMM 1999*, August 1999.
11. T. Erlebach and M. Rüegg, "Optimal Bandwidth Reservation in Hose-Model VPNs with Multi-Path Routing", *IEEE Infocom 2004*, March 2004.
12. J. A. Fingerhut, S. Suri, and J. S. Turner, "Designing Least-Cost Nonblocking Broadband Networks", *Journal of Algorithms*, 24(2), pp. 287-309, 1997.
13. M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, 1988.
14. W. D. Grover, *Mesh-based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*, Prentice Hall, 2003.
15. A. Gupta, A. Kumar, M. Pal, and T. Roughgarden, "Approximation via cost sharing: Simpler and better approximation algorithms for network design", *Journal of the ACM (JACM)*, vol. 54, issue 3, June 2007.

16. C. Harrelson, K. Hildrum, S. Rao, "A Polynomial-time Tree Decomposition to Minimize Congestion", *Symposium on Parallelism in Algorithms and Architectures (SPAA)*, June 2003.
17. S. Iyer, S. Bhattacharyya, N. Taft, C. Diot, "An approach to alleviate link overload as observed on an IP backbone", *IEEE Infocom 2003*, March 2003.
18. S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone", *IEEE Infocom 2003*, March 2003.
19. G. Italiano, R. Rastogi, and B. Yener, "Restoration Algorithms for VPN Hose Model", *IEEE Infocom 2002*, 2002.
20. M. Kodialam, T. V. Lakshman, and Sudipta Sengupta, "Efficient and Robust Routing of Highly Variable Traffic", *Third Workshop on Hot Topics in Networks (HotNets-III)*, November 2004.
21. M. Kodialam, T. V. Lakshman, J. B. Orlin, and Sudipta Sengupta, "A Versatile Scheme for Routing Highly Variable Traffic in Service Overlays and IP Backbones", *IEEE Infocom 2006*, April 2006.
22. M. Kodialam, T. V. Lakshman, J. B. Orlin, and Sudipta Sengupta, "Preconfiguring IP-over-Optical Networks to Handle Router Failures and Unpredictable Traffic", *IEEE Journal on Selected Areas in Communications (JSAC)*, Special Issue on Traffic Engineering for Multi-Layer Networks, June 2007.
23. M. Kodialam, T. V. Lakshman, and S. Sengupta, "Maximum Throughput Routing of Traffic in the Hose Model", *IEEE Infocom 2006*, April 2006.
24. M. Kodialam, T. V. Lakshman, and Sudipta Sengupta, "Throughput Guaranteed Restorable Routing Without Traffic Prediction", *IEEE ICNP 2006*, November 2006.
25. A. Kumar, R. Rastogi, A. Silberschatz , B. Yener, "Algorithms for provisioning VPNs in the hose model", *ACM SIGCOMM 2001*, August 2001.
26. C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental Study of Internet Stability and Backbone Failures", Proceedings of *29th International Symposium on Fault-Tolerant Computing (FTCS)*, Madison, Wisconsin, pp. 278-285, June 1999.
27. H. Liu and R. Zhang-Shen, "On Direct Routing in the Valiant Load-Balancing Architecture", *IEEE Globecom 2005*, November 2005.
28. A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, C. Diot, "Traffic Matrix Estimation: Existing Techniques and New Directions", *ACM SIGCOMM 2002*, August 2002.
29. H. Räcke, "Minimizing congestion in general networks", *43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.
30. R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, Morgan Kaufmann Publishers, 2002.
31. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
32. F. Shahrokhi and D. Matula, "The Maximum Concurrent Flow Problem", *Journal of ACM*, 37(2):318-334, 1990.
33. Sudipta Sengupta, *Efficient and Robust Routing of Highly Variable Traffic*, Ph.D. Thesis, Massachusetts Institute of Technology (MIT), December 2005.
34. N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP Topologies with Rocketfuel", *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 2-16, February 2004.
35. I. Stoica, D. Adkins, S. Zhuang, S. Shenker, S. Surana, "Internet Indirection Infrastructure", *ACM SIGCOMM 2002*, August 2002.
36. L. G. Valiant, "A scheme for fast parallel communication", *SIAM Journal on Computing*, 11(7), pp. 350-361, 1982.
37. Y. Zhang, M. Roughan, N. Duffield, A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads", *ACM SIGMETRICS 2003*, June 2003.
38. R. Zhang-Shen and N. McKeown "Designing a Predictable Internet Backbone Network", *Third Workshop on Hot Topics in Networks (HotNets-III)*, November 2004.
39. R. Zhang-Shen and N. McKeown, "Designing a Predictable Internet Backbone with Valiant Load-Balancing", *Thirteenth International Workshop on Quality of Service (IWQoS 2005)*, June 2005.

40. Y. Zhang, M. Roughan, C. Lund, and D. Donoho, “An Information-Theoretic Approach to Traffic Matrix Estimation”, *ACM SIGCOMM 2003*, August 2003.

Index