Efficient and Robust Routing of Highly Variable Traffic

Murali Kodialam   T. V. Lakshman   Sudipta Sengupta

Bell Laboratories, Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733, USA

ABSTRACT

We consider the following issues related to robust network routing in a highly dynamic and changing traffic environment: What network routing should an Internet Service Provider use so as to (i) accommodate users demanding "good" service while being unpredictable in the traffic that they would like to send to different destinations, (ii) minimize the amount of "overprovisioning" that needs to be done in the network in order to make "best effort networking better" without resorting to sophisticated traffic prediction and management mechanisms, (iii) operate the network efficiently with mostly static routing configurations and without dynamic routing adjustments to avoid congestion due to drastic changes in traffic flows between a network's ingress and egress routers. Achieving these goals has been difficult and has led to networks being very much overprovisioned in order to avoid the management complexity of implementing traffic management schemes that adapt network routing to changed traffic demands.

In this paper, we propose a simple network routing scheme that is not much more complex to implement than shortest path routing with the following properties: (i) it effectively handles all traffic patterns permissible within the capacity constraints of ingress-egress links, (ii) it avoids congestion without requiring dynamic reconfiguration of routing parameters (such as link weights), and (ii) it is bandwidth efficient despite the ability to handle all traffic matrices. We argue that the routing scheme we propose is very effective in avoiding network congestion under extreme traffic variability while being static in the routing configuration and parsimonious in its "overprovisioning".

1. INTRODUCTION

In this paper, we propose a simple static routing scheme that is robust to extreme traffic fluctuations without requiring significant network overprovisioning. Specifically, the method we propose has the following properties: (i) It can handle any traffic pattern permissible within the constraints imposed by the network's edge-link capacities, (ii) It avoids network congestion under high traffic variability without requiring dynamic link weight or routing policy adjustments, (iii) Its capacity requirements are close to that needed to accommodate one "upper bound" traffic pattern even though it can handle all possible traffic patterns subject to ingress-egress capacity constraints. The ability to handle large traffic variations with a fixed routing scheme can greatly simplify network operation and our scheme is effective because these goals can be achieved without incurring high overheads in capacity costs.

1.1 Causes for Traffic Variation

Extreme network traffic fluctuations can happen for a variety of reasons. Consider a large Internet service provider exchanging traffic with several other providers. Typically, the traffic exchange between carriers is specified by total traffic volumes over long time periods and possibly a peak rate limit (usually just determined by physical link capacities). The actual distribution of traffic entering at an ingress to the various network egresses is not known and can change over time. This is because the distribution is determined by many factors such as intrinsic changes of traffic to different destination prefixes and by routing changes either made locally by the carrier or due to changes made in other ASes over which the carrier has no control. Intrinsic changes in traffic distribution can be caused by many factors such as the sudden appearance of flash crowds responding to special events. An example of local routing changes that can affect the traffic distribution is IGP weight changes combined with hot-potato routing that can change the network egress that traffic destined to a set of prefixes would choose. Another example is MED changes in BGP. While local routing changes are under a carrier's control and hence change traffic patterns only at planned instants, unpredictable traffic shifts can happen when routing changes in other ASes affect downstream ASes. A recent study of the effects of the prevalent hot-potato routing [1] shows that IGP weight changes (which can be due to new links being added, maintenance, traffic engineering, etc.) in an AS can cause significant shifts in traffic patterns. Example are shown where changes in IGP costs can affect the BGP route for 40% of the prefixes, and Netflow measurements are shown to indicate that the affected prefixes can account for upto 35% of the traffic. This indicates that significant shifts in traffic may happen at a carrier due to changes elsewhere in the network.

Another example application where the traffic matrix is unknown is the provisioning of network-based VPN services to enterprise customers. Here, customers do not know their traffic matrices and only specify to the carrier the total traffic volume and the peak rate. It is the carrier's task to transport all of the offered VPN traffic to the network and carry them without introducing too much delay. Again, the proposed scheme is well-suited to the needs of this application. The proposed routing is applicable in many other scenarios where traffic variations can be extreme and the traffic matrix is not known a priori, for example, in grid computing. Note that the focus of this paper is for the case when the traffic matrix is unknown. The case when the matrix is known has received considerable research attention and is not considered in this paper.

1.2 Preferred Routing Characteristics

To provide good service when traffic patterns can change uncontrollably, carriers must either quickly and repeatedly adapt their intra-domain routing to avoid network congestion or must have sufficient capacity set aside a priori to accommodate the different traffic patterns that can occur without resorting to routing changes. Service providers prefer to avoid frequent intra-domain routing changes due to operational complexity and costs, and due to the risk of network instability if link metric changes are not implemented correctly. Moreover, changes in one AS in the BGP application above may cause cascading traffic changes in other ASes affecting the overall stability of many Internet paths. The trade-off in avoiding routing changes is the significant capacity overprovisioning that must be done to accommodate changing traffic patterns while keeping the routing fixed. Ideally, providers would like to use a fixed routing scheme that does not require traffic dependent dynamic adaptation of configuration parameters and which is parsimonious in its capacity needs.

1.3 Proposed Routing Strategy

The scheme that we propose is based on the idea of replacing shortest path IGP routing within a carrier's domain by a modified routing scheme that routes traffic to the destination after ensuring that it passes through a pre-determined intermediate node also in the carrier's domain. (The assignment of an intermediate node can be made at the flow level to avoid packet resequencing issues.) Note that the egress nodes are still chosen based on BGP-determined AS paths and auxiliary carrier routing policies such as hot potato routing. Our scheme only changes the IGP path selection of direct shortest paths to one which passes through a apriori assigned intermediate node. In MPLS networks, this routing through a pre-determined intermediate node can be accomplished using a pre-configured set of MPLS LSPs between the each ingress and a chosen set of intermediate nodes to which flows are assigned according to specified probabilities. In pure IP networks, this routing can be accomplished by tunneling packets to the pre-determined intermediate node first. This routing with pre-determined selection of an intermediate node is sufficient to handle all traffic patterns that are permissible subject to edge-link capacity constraints. Moreover, routing adaptations are not needed when the traffic matrix changes and the scheme is bandwidth efficient.

1.4 Suitability for IP-over-Optical Networks

Routing in IP-over-Optical networks (where routers are interconnected over a switched optical backbone) needs to make a compromise between keeping traffic at the optical layer (for network cost reasons) and using intermediate routers for packet grooming in order to achieve efficient statistical multiplexing of data traffic. The proposed scheme, when applied to IP-over-Optical networks, routes packets in the optical layer with *packet grooming at one intermediate router* only *and* provides the desirable statistical multiplexing properties of packet switching with highly variable traffic.

2. MODELING VARIABILITY IN NETWORK TRAFFIC

We assume that we are given a network $G = (N, E)$ with nodes N and (directed) edges E where each node in the network can be an ingress-egress point. Let $|N| = n$ and $|E| = m$. We let (i, j) represent a directed link in the network from node i to node j. To simplify the notation, we will also refer to a link by e instead of (i, j).

The total amount of traffic that enters (leaves) an ingress (egress) node in the network is bounded by the total capacity of all external ingress links (e.g., line cards to customer networks or other carriers) at that node. Denote the upper bounds on the total amount of traffic entering and leaving the network at node i by R_i and C_i respectively (Figure 1).

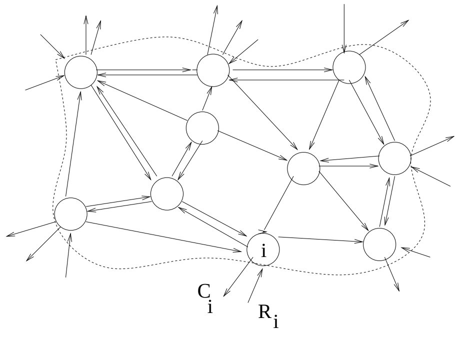

Figure 1: Traffic Model

The point-to-point matrix for the traffic in the network is thus constrained by these ingress-egress link 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 allowable traffic matrix $T = [t_{ij}]$ for the network must obey $\sum_{j:j\neq i}^{n} t_{ij} = R_i$ and $\sum_{j:j\neq i}^{n} t_{ji} = C_i$ for all $i \in N$.

We briefly argue why it is sufficient to consider equality (and not $\leq$) in the above constraints. Any matrix T' whose any row or column sums up to a value less than the given bounds can be transformed to a matrix T (with equality in the constraints) by addition of a matrix T'' with non-negative (non-diagonal) entries, i.e., $T = T' + T''$. Thus, any routing scheme that routes T can route T' also.

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 $\mathcal{T}(\mathcal{R}, \mathcal{C})$, that is

$$\mathcal{T}(\mathcal{R}, \mathcal{C}) = \{[t_{ij}] : \sum_{j\neq i} t_{ij} = R_i \text{ and } \sum_{j\neq i} t_{ji} = C_i \ \ \forall\ i\}$$

Note that the traffic distribution T could be any matrix in $\mathcal{T}(\mathcal{R}, \mathcal{C})$ and could change over time. We would like to have a routing architecture that does not make any assumptions about T apart from the fact that it is partially specified by row and column sum bounds. In this context, we investigate the following question:

Does there exist a routing strategy that (i) can route every matrix in $\mathcal{T}(\mathcal{R}, \mathcal{C})$, (ii) does not require reconfiguration of existing connections, i.e., is oblivious to changes in the traffic matrix T as long as it belongs to $\mathcal{T}(\mathcal{R}, \mathcal{C})$, and (iii) is bandwidth efficient?

By bandwidth efficiency, we mean that the routing scheme should (i) not use much more bandwidth than that for routing any single matrix in $\mathcal{T}(\mathcal{R}, \mathcal{C})$ and (ii) use significantly less bandwidth than the (obvious) expensive strategy of provisioning $\min(R_i, C_j)$ amount of demand from node i to node j. In the next section, we describe a routing architecture that meets the above design requirements.

2.1 Related Work

The hose model was proposed by Duffield et al. [10] as a method for specifying the bandwidth requirements of a Virtual Private Network (VPN). Given per-link costs, the problem of minimum cost capacity reservation under the hose model has been considered under the tree routing [11], single-path routing [11], and multi-path routing [12] models. These results, with the exception of [12], are all for the uncapacitated case where link capacities are infinite. For finite link capacities, the problem of finding a feasible tree routing or single-path routing is shown to be $\mathcal{NP}$-hard in [11].

In all of the above models, even though the paths are fixed apriori and do not depend on the traffic matrix, their bandwidths change with variations in the traffic matrix. In an IP/MPLS or IP-over-Optical network, *deployment of any of the above routing models necessitates reconfiguration of the provisioned paths in response to traffic variations.* This makes network routing less stable and predictable. *In contrast, for the routing scheme we propose, both the paths and their bandwidth are fixed apriori and do not need to be changed as traffic patterns change over time* (subject to the ingress-egress capacity constraints).

Recently, a routing strategy that initially routes packets to a randomly chosen output port in a switch and then routes to the true destination has been proposed [2] as an effective scheme for avoiding scheduling bottlenecks in high-speed input-buffered switches. The scheme has been extended to apply to network-wide routing [3] [4] [5] [6] [8]. The use of randomization in network routing was first proposed by [9] and subsequently studied extensively.

Our current work differs in many ways from and is complementary to [4] where the authors consider the impact of arbitrary (logical) link and node failures for the special case of routing with *equal splits to all nodes.* We propose a generalized scheme with possibly unequal split ratios and considers the problem of minimum bandwidth (physical) routing under router node capacity constraints. In [7], we address the problem of routing under given link capacities so as to maximize throughput. In [3], we consider making the scheme resilient to router node and optical layer link failures in the context of IP-over-Optical networks.

3. PROPOSED ROUTING SCHEME

In this section, we describe a routing scheme that allows the network to meet arbitrary (and possibly rapidly changing) traffic demands without sophisticated traffic engineering mechanisms or additional network signaling. In fact, the scheme does not even require the network to detect changes in the traffic distribution. The only assumption about the traffic is the limits imposed by the total capacity of all line cards that connect to external interfaces at network edges.

The proposed routing strategy operates in two phases:

- **Phase 1:** A pre-determined 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.

This is illustrated in Figure 2. A simple method of implementing this routing scheme in the network is to form *fixed bandwidth tunnels between the nodes.* In order to differentiate the tunnels carrying Phase 1 and Phase 2 traffic, we will refer to these tunnels as Phase 1 and Phase 2 tunnels respectively. The critical reason that the two phase routing strategy works is that the *bandwidth required for these tunnels only depends on R and C values and not on the (unknown) individual entries in the traffic matrix.*

Note that the traffic split ratios $\alpha_1, \alpha_2, \ldots, \alpha_n$ in Phase 1 of the scheme are such that $\sum_{i=1}^{n} \alpha_i = 1$. Let us elaborate on the routing procedure. 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 demand from node i to node j as a result of Phase 1 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 $\alpha_i t_{kj}$ as long as all traffic is initially split without regard to the final destination. Thus, the maximum traffic that needs to be routed from node i to node j during Phase 2 is $\sum_{k \in N} \alpha_i t_{kj} = \alpha_i C_j$. Thus, the traffic demand from node i to node j during Phase 2 is $\alpha_i C_j$.

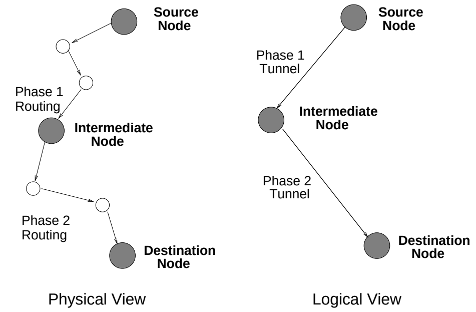

Figure 2: Phase 1 and Phase 2 Routing for the Proposed Scheme

Thus, 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 $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$. Three 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 pre-determined traffic split ratios and *does not depend* on the specific traffic matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$.

Property 2 (Provisioned Capacity is Traffic Matrix Independent): The total demand between nodes i and j as a result of routing in Phases 1 and 2 is $t'_{ij} = \alpha_j R_i + \alpha_i C_j$ and does not depend on the specific matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$ but only on the row and column sum bounds that constrain T (i.e., define the set $\mathcal{T}(\mathcal{R}, \mathcal{C})$).

Property 3 (Complete Utilization of Provisioned Capacity): For each matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$, the routing scheme can completely utilize the associated point-to-point demands in Phases 1 and 2.

Property 2 implies that the scheme handles variability in traffic matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$ by effectively routing a transformed matrix $T' = [t'_{ij}]$ that depends only on the row and column sum bounds and the distribution ratios $\alpha_1, \alpha_2, \ldots, \alpha_n$, and not on the specific matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$. This is what makes the routing scheme oblivious to changes in the traffic distribution.

The fact that the traffic distribution obeys the row (column) sum bounds can be enforced in a couple of ways. By making the row (column) sum bounds equal to the sum of the line card capacities that connect to external interfaces at a node, the constraint is enforced in a hard manner (at the physical layer). Alternatively, a DiffServ type policing scheme can rate limit the total traffic that enters the network at each ingress node and guarantee that each node is not over-subscribed.

We highlight below the main aspects of the novelty of the proposed scheme:

1. Routing decisions at each source node during Phase 1 are local do not require any network-wide state information (e.g., how the traffic at other peering points is varying). Routing decisions during Phase 2 are based on the packet destination only as with current IP network routing.

2. The network can meet any traffic distribution as long as the ingress/egress points are not over-subscribed. Congestion can be avoided by either hard rate guarantees of line cards connecting to other carriers, or by implementing a DiffServ type policing scheme for rate limiting the traffic entering the network at a node.

3. The routing scheme is oblivious of and robust to any changes in the traffic distribution. Providing end-to-end bandwidth guarantees does not require any reconfiguration of the network in real-time.

From a network operations perspective, we outline below the phases involved in the routing architecture implementation.

Network Setup:

1. Compute row (column) bounds R_i (C_i) using inter-AS peering agreements and/or rates of line cards at each node connecting to other carriers.

2. Compute traffic distribution ratios $\alpha_1, \alpha_2, \ldots, \alpha_n$ (an algorithm for this that optimizes the required network bandwidth is described later).

3. For each node pair i, j, provision two connections (MPLS LSPs, IP tunnels, or optical layer circuits, as the case may be) from i to j of bandwidth, one for Phase 1 of bandwidth $\alpha_j R_i$, and the other for Phase 2 of bandwidth $\alpha_i C_j$.

Network Routing:

1. Traffic is routed in accordance with Phases 1 and 2 (described earlier) that require only local operations at source and intermediate nodes.

2. DiffServ type policing mechanism is used to rate limit the total traffic that enters the network at each node.

3. If bounds R_i (C_i) change as a result of new peering agreements or modifications to existing ones, the bandwidth of the LSPs (or IP tunnels) for routing during Phases 1 and 2 can be adjusted accordingly (after possible re-optimization of the distribution ratios α_i's.)

The traffic split ratios can be generalized to depend on *source and/or destination nodes* of the traffic also. We consider this in Section 3.2

3.1 Capacity Minimization and Linear Programming Formulation

We outline a linear programming formulations for minimum bandwidth routing for the proposed scheme under node capacity constraints. By bandwidth, we mean the total router port usage across all nodes in the network. Each node i has capacity u_i for the total traffic going through it that models the chassis capacity of the router at that node. We model the maximum router chassis capacity only; the line cards for the router at a node will be populated (up to the chassis capacity) in accordance with the traffic through that node.

Adopting the standard multi-commodity flow terminology [13], we use the term commodity to indicate the flow between a source and a destination. We use k to index the commodities. The source node for commodity k will be denoted by $s(k)$ and the destination node by $d(k)$. We use $x^k(e)$ to denote the amount of flow of commodity k on link e in the network. The sets of incoming and outgoing edges at node i are denoted by $\delta^-(i)$ and $\delta^+(i)$ respectively. There are two sets of decision variables, the fraction of traffic that will be routed to node i in the first phase denoted by α_i, and the flows on link e for commodity k denoted by $x^k(e)$. Note that the demand for commodity k will be given by $\alpha_{s(k)}C_{d(k)} + \alpha_{d(k)}R_{s(k)}$.

$$\min \textstyle\sum_{e\in E}\sum_k x^k(e)$$

subject to

$$\sum_{e\in\delta^-(i)} x^k(e) = \sum_{e\in\delta^+(i)} x^k(e) \qquad \forall\ i \neq s(k), d(k),\ \ \forall\ k \tag{1}$$

$$\sum_{e\in\delta^+(i)} x^k(e) = \alpha_{s(k)}C_{d(k)} + \alpha_{d(k)}R_{s(k)} \qquad i = s(k),\ \ \forall\ k \tag{2}$$

$$\sum_k \sum_{e\in\delta^+(i)} x^k(e) + C_i \le u_i \ \ \forall\ i \tag{3}$$

$$\sum_i \alpha_i = 1 \tag{4}$$

3.2 Generalized Traffic Split Ratios

The traffic split ratios α_i can be generalized to depend on *source or destination nodes* of the traffic, or both. We discuss the latter version here.

Suppose that a fraction α_k^{ij} of the traffic that originates at node i whose destination is node j is routed to node k in the intermediate stage. We now would like to compute the capacity that is needed between nodes i and j in the first and second phase. Let the current traffic matrix be $T = [t_{ij}] \in \mathcal{T}(\mathcal{R}, \mathcal{C})$. In the first phase, the capacity needed between nodes i and j is

$$\sum_k \alpha_j^{ik} t_{ik} \le \max_k \alpha_j^{ik} \sum_k t_{ik} = \max_k \alpha_j^{ik} R_i$$

For the second phase, the capacity needed between nodes i and j is given by

$$\sum_k \alpha_i^{kj} t_{kj} \le \max_k \alpha_i^{kj} \sum_k t_{kj} = \max_k \alpha_i^{kj} C_j$$

Therefore the total capacity needed between nodes i and j in Phase 1 and Phase 2 together is

$$C_{ij} \ge \alpha_j^{ik} R_i + \alpha_i^{mj} C_j \ \ \forall\ k\ \ \forall\ m$$

Note that the constraint above is linear and independent of the individual entries in the traffic matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$ and is only dependent on the row and column sums and the traffic split ratios. Hence, it can be easily accommodated into the linear programming formulation of Section 3.1.

3.3 Throughput Maximization

Given a network with *link capacities* and constraints R_i, C_i on the ingress/egress traffic, the problem of routing under the proposed scheme so as to minimize the maximum utilization of any link in the network can also be formulated as a linear program [7]. (The utilization of a link is defined as the traffic on the link divided by its capacity.) Let $\lambda \cdot \mathcal{T}(\mathcal{R}, \mathcal{C})$ denote the set of all traffic matrices in $\mathcal{T}(\mathcal{R}, \mathcal{C})$ with their entries multiplied by λ. Then, this problem is equivalent to finding the maximum multiplier λ (throughput) such that all matrices in $\lambda \cdot \mathcal{T}(\mathcal{R}, \mathcal{C})$ can be routed. A fast combinatorial fully polynomial time approximation scheme (FPTAS) for this problem is also presented in [7].

4. CAPACITY EFFECTIVENESS EVALUATION

Next, we evaluate the capacity performance of the proposed routing scheme. For this purpose, we need to define the *bandwidth efficiency* of a routing scheme and compare it with that of the best possible scheme in the class of all schemes that route all matrices in $\mathcal{T}(\mathcal{R}, \mathcal{C})$.

Definition (Bandwidth Efficiency): Denote the vector of node capacities by $u = (u_1, u_2, \ldots, u_n)$ Suppose that the minimum possible bandwidth usage admitted by *any routing scheme* under given node capacities u is $\hat{C}_u$. Denote by C_u the same quantity for our proposed scheme (this is computed by the linear program of Section 3.1). Clearly, $\hat{C}_u \leq C_u$. We define the bandwidth efficiency of our routing scheme by the quantity $\hat{C}_u/C_u$ (≤ 1).

The value $\hat{C}_u$ is hard to compute. However, suppose that we take any single matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$ and compute the minimum bandwidth $C_u(T)$ (using a multicommodity flow formulation [13]) required for routing this single matrix under given node capacities u. Then, $C_u(T) \leq \hat{C}_u$, and hence

$$\frac{C_u(T)}{C_u} \leq \frac{\hat{C}_u}{C_u} \leq 1$$

Thus, for any traffic matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$, the quantity $C_u(T)/C_u$ is a lower bound on the bandwidth efficiency of our routing scheme. To obtain a tight lower bound, we would like to identify a matrix $T \in \mathcal{T}(\mathcal{R}, \mathcal{C})$ for which $C_u(T)$ is maximum. This matrix T is hard to compute; we use a heuristic approach to find a matrix that gives tight lower bounds. The details are omitted for lack of space.

For the results presented here, the R_i's and C_i's are assumed to be equal and normalized to 1, i.e., $R_i = C_i = 1$ for all i. All the router capacities u_i are also identical and denoted by u_R. Below a minimum value of the router capacity u_R, the routing problem will be infeasible. Above a certain value of the router capacity, the optimal objective function for the routing problem will remain the same. The bandwidth efficiency is plotted in this feasible range of node capacities.

We consider two network topologies – (i) a 15-node network with 28 bidirectional links, and (ii) a 20-node network with 33 bidirectional links. These topologies are representative of US carrier backbone networks in their size range. For the results, we solved the linear program using the commercially available linear programming package `cplex`. The bandwidth efficiency of our scheme, as defined above, is plotted for the 15-node and 20-node networks in Figure 3.

For the 15-node topology, routing under the proposed scheme becomes feasible at $u_R = 2.335$. With increasing value of u_R, the bandwidth efficiency value flattens out at around 96% (for $u_R = 2.7$), indicating that the bandwidth usage of the our scheme is very close to the best possible.

For the 20-node topology, routing under the scheme becomes feasible at $u_R = 2.595$. With increasing value of u_R, the the bandwidth efficiency value flattens out at around 94% (for $u_R = 2.8$), again indicating that our scheme is very close to the best possible in terms of bandwidth usage for routing all matrices in $\mathcal{T}(\mathcal{R}, \mathcal{C})$.

The above results point to the important conclusion that our proposed routing scheme is able to route efficiently with traffic uncertainty (under the defined traffic variation model) with router port usage not significantly higher (less than 10%) than that for a single matrix chosen for the traffic distribution. Note that under any routing scheme that the network service provider considers, capacity needs to be provisioned for any achievable traffic matrix. Equivalently, the bandwidth efficiency of the scheme is *very close to the best possible* for routing all matrices under the defined traffic model.

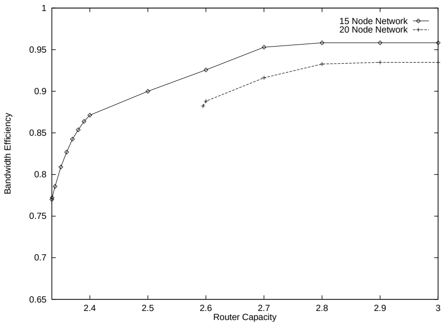

Figure 3: Bandwidth Efficiency of Proposed Scheme

5. REFERENCES

[1] R. Teixeira, A. Shaikh, T. Griffin, J. Rexford, "Dynamics of Hot-Potato Routing in IP Networks", *ACM SIGMETRICS 2004*, June 2004.

[2] C. -S. Chang, D. -S. Lee, and Y. -S. Jou, "Load balanced Birkhoff-von-Neumann switches, Part I: one-stage buffering", *Computer Communications*, vol. 25, pp. 611-622, 2002.

[3] M. Kodialam, T. V. Lakshman, and S. Sengupta, "Efficient, Robust Routing in Highly Dynamic Environments", Stanford Workshop on Load-Balancing, May 2004.

[4] R. Zhang-Shen and N. McKeown, "Designing a Predictable Internet Backbone", Stanford Workshop on Load-Balancing, May 2004.

[5] R. Zhang-Shen and N. McKeown "Designing a Predictable Internet Backbone Network", Third Workshop on Hot Topics in Networks (HotNets-III), November 2004.

[6] M. Zirngibl, D. Stiliadis, P. Winzer, H. Nagesh, V. Poosala, "New Networking Architectures and Technologies", Stanford Workshop on Load-Balancing, May 2004.

[7] M. Kodialam, T. V. Lakshman, and S. Sengupta, "Guaranteeing Predictable Performance to Unpredictable Traffic: Routing and Throughput Maximization", Bell Laboratories Technical Report, ITD-04-45850M, October 14, 2004.

[8] H. Nagesh, V. Poosala, V.P. Kumar, "NetSwitch: Load Balanced Data Optical Architecture", Personal Communication, August 2004.

[9] L. G. Valiant, "A scheme for fast parallel communication", *SIAM Journal on Computing*, 11(7), pp. 350-361, 1982.

[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 networks", *ACM SIGCOMM 1999*, August 1999.

[11] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, B. Yener, "Provisioning a Virtual Private Network: A Network Design Problem for Multicommodity Flow", *ACM Symposium on Theory of Computing (STOC) 2001*, July 2001.

[12] T. Erlebach and M. Ruegg, "Optimal Bandwidth Reservation in Hose-Model VPNs with Multi-Path Routing", *IEEE Infocom 2004*, March 2004.

[13] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, February 1993.