Achieving IP Routing Stability with Optical Bypass

Mohit Chamania[a], Marcel Caria[a], Admela Jukan[a]

[a] *Technische Universität Carolo-Wilhelmina zu Braunschweig, Germany, {chamania, caria, jukan}@ida.ing.tu-bs.de*

Abstract

The phenomenal growth of the Internet coupled with the emergence of new QoS-aware services in the network has put an enormous strain on current networks. Research efforts towards optimizing the performance of IP networks have been focused on increasing the utilization of the network while minimizing additional resources used. Many such paradigms developed under the umbrella of Traffic and Network Engineering can lead to frequent and significant routing changes in the network when used to tackle short-lived traffic churns, and are therefore not commonly used by network providers. In this work, we present a new network engineering paradigm targeted towards handling short-lived traffic bursts which ensures that *IP routing remains unchanged* at the cost of marginally higher overall resource usage. We present an ideal Integer Linear Programming (ILP) based optimization problem and present its adaptations to real networks where the actual end-to-end traffic is not known. The latter is of significant practical interest as determination of the IP traffic matrix is non-trivial. Our results are promising and show that the proposed model for IP-optical interoperability model can ensure stable IP operation at a marginally higher resource cost, even when the IP traffic matrix is not known.

Key words: Network Engineering, multi-layer networks, hybrid networks

1. Introduction

The Internet has not only witnessed a phenomenal growth in traffic but has also seen a steep rise in the number of service-aware applications. While IP was not designed to facilitate QoS in the network, the widespread deployment of IP in the global network has led to the development of mechanisms which attempt to optimize performance of the IP network to support new service aware applications. Network providers are therefore looking for solutions which can optimize performance in the IP network (implying high utilization of resources) while maintaining operational stability. Current resource optimization solutions

A short summarized version of this paper was presented at the Third Symposium on Advanced Networks and Telecom Systems (ANTS) 2009 in New Delhi, India, in December 2009.

for IP networks, broadly classified under Traffic and Network Engineering, do not address issues pertaining to operational stability of IP networks, and are therefore implemented sparingly in commercial networks. Network providers instead use network overprovisioning paradigms to ensure smooth operation in networks. A point in case is not only the Interent2 network [1], but also manifold large commercial ISPs where network links are provisioned so that the link utilization at peak network loads is limited to about 30% of the link capacity, or less.

Current techniques for network optimization focus solely on optimal resource utilization in the network. Traffic Engineering (TE)techniques traditionally compute IP routing via a min-max optimization where the maximum link utilization in the network is minimized. While significantly more effective than traditional shortest path routing paradigms, TE techniques are limited by the total installed capacity in the network, and cannot guarantee stable (congestion free) network operation in case of link failures or sudden traffic spikes. Network Engineering (NE) techniques, on the other hand, are designed to ensure congestion-free operation in combined IP and optical ("hybrid") networks, and are formulated as an optimization problem which minimizes the new link capacity added to the IP network while ensuring that there is no congestion in the IP layer. NE operations do lead to changes in the IP network topology and IP routing, and therefore, current service providers in spite of having hybrid network capabilities, have preferred overprovisioning of IP links over dynamic reconfigurability to avoid frequent routing changes. The premium on stability is governed not only by the routing disruption caused in the IP control plane but also by the significant re-configuration required in the IP management plane. Change in IP routing can lead to re-routing of significant number of flows in the network, which in turn leads to significant reconfiguration of the monitoring systems installed to monitor Service Level Agreement(SLA) compliance and help with fault detection in the network. The change in routing also leads to significant re-configuration of the Event Correlation Systems which use monitoring measurements from different services and diverse sites in the network to trigger alarms. The re-configuration of the monitoring services and the event correlation database is a tedious and error-prone operation, which is why network operators tend to avoid IP topology reconfiguration techniques.

Therefore, for short-lived events like high bandwidth application flows or temporary failures causing congestion in the network, it might be more appropriate to employ mechanisms which ensure that routing is not affected significantly in the IP network, albeit at a higher cost of resources used. Stability of routing would ensure minimal reconfiguration of the event correlation database and minimal service disruptions in the data layer, and is better suited for short-term disruptions which can occur frequently. Since, current network engineering approaches are therefore more suited for permanent congestion events, a new approach is required to handle temporary and unexpected congestion events in the network. It is our proposition that any approach attempting to tackle short term congestion/failure events should be required to cause minimal or no rout-

ing changes in the IP network, and affect as few aggregate flows* as possible. These two simple and basic requirements ensure that the effect of any NE operation minimizes the reconfiguration effort in the Network Management Systems (NMS), and ensures operational stability while guaranteeing congestion-free operation in the IP layer.

In this paper, we present a new IP-optical network engineering paradigm which is governed by the abovementioned features and is ideal for dealing with short-lived congestion events in the network. We coin the term "optical bypass" to refer to the choice of dynamic circuit services enabled by optical transmission and switching, such as those that can be provided by TDM networks, WDM networks or connection-oriented packet switched networks with wireline or pseudowire emulation capabilities, such as MPLS-TP and carrier Ethernet. However, the choice of packet switching technology is assumed to be exclusively IP. The preliminary optimization model for our approach was presented in [2], and we extended the same to account for incomplete knowledge of the traffic matrix in IP networks in [3]. In this paper, we present additional extensions to our optimization model using a modified version of the so-called *tomogravity* model [4] which is used for traffic estimation. Our approach with bypass based routing ensures that there are no significant routing changes in the IP network while guaranteeing stability of operation. Alternate optimization models presented in this paper also show that our mechanism can be employed even when traffic matrix information is not available in the network.

The rest of the paper is organized as follows: In section 2, we present a brief overview of the current network engineering solutions and present the basic outline of our solution. In section 3, we present the required network capabilities of the ideal bypass-based routing solution and present an optimization model for the same. Section 4 presents various techniques to deploy the bypass based routing mechanisms in networks when traffic matrix information is not available. Section 5 presents a numerical analysis of the performance of the various techniques proposed while section 6 concludes the paper.

2. Background and Motivation

In this section we describe the evolution of hybrid IP-over-circuit networking paradigms and present the motivation behind our approach.

2.1. LambdaStation

The LambdaStation Project [5] implemented a multi-layer network architecture wherein the IP network was used to route data end-to-end over static IP links, but end-user applications could decide to route long-lived high-bandwidth

*Definition of a "flow" in our paper: As it is not always feasible to measure flows on an application level granularity, we define an aggregate flow between two routers in the network as all application flows entering from the source router and exiting from the destination router.

flows over dynamic end-to-end circuits. In the LambdaStation architecture, applications were provided with an API which directed the network to switch a specific *application flow* (identified by a 5-tuple: source IP, dest. IP, source port, dest port, transport protocol) over an end-to-end circuit. A centralized controller was used which scheduled the creation and tear-down of an end-to-end circuit based on the API calls by user applications, and also configured the routing of individual application flows onto the end-to-end circuits using Policy Based Routing mechanisms [6]. While the proposed mechanism has been greatly successful in the scientific research networks, where users are empowered to schedule circuit setup/tear-down, it has been only marginally discussed for applications in commercial ISP networks. In the commercial environments, most of the focus was on methods based on flow classification, as discussed next.

2.2. Flow Classification

To avoid the user-based decision for selecting a circuit service for a specific application flow, the next generation of multi-layer networks developed algorithms which attempted to identify high-bandwidth long-lived flows, and would trigger end-to-end circuits for these flows. For commercial ISPs, this approach presents an advantage over the LambdaStation approach as the network operators had better control over the flows that are to be switched and routed onto circuits. Advanced flow monitoring services were introduced and different approaches were proposed to enable flow classification and control switching of end-to-end bypasses, such as [7], [8] and [9] . While these approaches were a step towards more commercial implementations, their effectiveness is directly related to the efficiency of the flow classification mechanism itself. Also, given that only a very few individual application flows are likely to be large enough to cause congestion in commercial networks, this approach is likely to have limited practical implications for commercial networks.

2.3. Network Engineering

Network Engineering [11] is a relatively new approach designed for multi-layer networks which attempted to selectively add/remove capacity from the IP network by employing dynamic circuits established between routers in the transport layer. The newly established circuits were then advertised as new IP links and routing in the network would adapt to the new IP topology. Traditionally, network engineering problems were modeled as optimization problems which attempted to minimize the circuit capacity installed in the network while eliminating congestion. At first glance, this approach addressed all the problems of its predecessors: however, network engineering approaches have not been deployed extensively in current networks. Network Engineering operations lead to change in the IP network topology, which cause significant routing changes in the network and current service providers, in spite of having network engineering capabilities, have preferred over-provisioning of IP links to avoid frequent routing changes. In fact, Network Engineering practice has been limited to infrequent implementation between subsequent network planning cycles to tackle long-term traffic increases.

2.4. Our Approach

It is our proposition that traditional NE solutions are suitable for addressing long-term traffic increases, while a network engineering approach is needed to tackle short term congestion/failure events. In doing so, the new approach must minimize or cause no routing changes in the IP network, and should affect routing of as few aggregate flows as possible. By minimizing the changes in IP routing and rerouting as few aggregate flows as possible, the re-configuration effort in the NMS is minimized, which is crucial to the stability when handling short lived flows. The new network engineering paradigm must also ensure that any attempt to reduce congestion on a link in the IP network should not lead to increase in traffic on any other existing link in network so that that the re-configuration does not increase the probability of congestion at any other remote site in the network.

Some possible approaches to address these constraints can be:

1) Restricting network engineering operations to temporary addition of capacity on existing links (a modified idea to the common practice to permanently increase the link capacity as soon as the utilization reaches a certain threshold), and/or

2) Exclusively using end-to-end circuits between source and destination routers to ensure that routing of other flows is not disturbed (a modified idea to the common practice of application APIs and flow classification tools which detect single application flows from source to destination).

In this paper however, we move even further in the ambition to provide a generic framework and solution to the problem of IP-optical integration, and present an approach whose solution encompasses, but is not restricted to, the solution set of the aforementioned options. Our proposal, termed as bypass-based routing, proposes to use dynamic circuits to create *bypasses* in the IP network. Unlike traditional network engineering approaches, bypasses are not advertised in the IP network, and only the router at the bypass ingress and egress are aware of the existence of the bypass. The ingress router, typically upstream from a congested node/link, can re-route specific aggregate flows onto the established bypass to the egress router, downstream from the congestion site, in order to reduce traffic in the congested area. However, the routers at the bypass ingress and egress must belong to the original routing path of the aggregate flow in order to ensure that the *bypassing* of the aggregate flow does not lead to increase in traffic at other sites in the network. The solution is fully compatible with the destination-based IP routing and does not require policy based router configuration.

To illustrate our proposal, consider the example network presented in Fig. 1: the link $R_1 - R_2$ is congested, and a bypass is established from R_1 to R_3 to divert traffic from this link. By our constraints, the bypass from R_1 to R_3 can only be used to reroute flows which have R_1 and R_3 in their original routing path (here the flow from $R_1 - R_3$) and therefore cannot be used to re-route traffic from R_1 to R_2. This constraint is used to ensure that no existing links in the network observe an increase in traffic. For instance, if the traffic from R_1

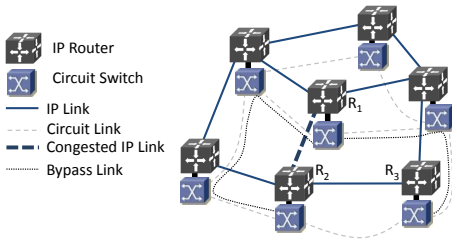

Figure 1: A multi-layer topology with congested links demonstrating the use of the proposed bypass mechanism

to R_2 was bypassed from R_1 to R_3 , the link from R_3 to R_2 would experience an increase in traffic, which is not desired. Note here that a bypass can also be established end-to-end or parallel to an existing link, but must still fulfill the criteria of the bypass-based routing mechanism.

We believe that the aforementioned characteristics ensure that the proposed solution meets the requirements of our ideal IP-optical networking solution targeted for short term events. By not advertising the bypass in the IP layer, we ensure that IP routing is not affected due to the establishment or the subsequent teardown of bypasses in the network. This mechanism also ensures that rerouting of a large number of flows in the IP network is avoided. We also constrain the bypassing of aggregate flows to bypasses which have end-points on the original routing path. With this constraint, we ensure that the aggregate flow follows the original routing path upstream and downstream from the bypass, and hence does not lead to *traffic increase* in any other section of the network. Given that the egress of the bypass lies on the original routing path, we observe that the egress does not have to specify specific routing rules for the incoming bypass flow, and this flow is automatically routed along the original routing path. Finally, a bypass can only be torn down after a fixed time interval Δ_T and only when we are certain that the tear-down of the bypass will not lead to congestion in the bypassed links.

In the next sections (Section 3 and 4) we present the architecture, assumptions and the mathematical formulations used for the bypass based routing mechanisms. In a separate section, we address a challenging case when IP traffic matrix is not known.

3. Mathematical Model for Bypass Based Routing

In order to formulate the objectives discussed previously, our model requires four major capabilities, namely:

- Network topology and routing information
- Bypass establishment service
- Traffic matrix information

- Failure/congestion identification

Network Topology and Routing Information

In order to develop models for bypass computation, it is assumed that we have complete topology information for both the IP network and the underlying transport network. This information can be easily obtained via the network management system, or by obtained by independently probing each router/switch in the IP and the transport network using SNMP. We assume that the circuit network is defined by the graph $G^p(V^p, E^p)$, with vertices $v_i^p \in V^p$ and edges $e_{ij}^p \in E^p$, and the IP network topology is given by the graph $G^l(V^l, E^l)$, with vertices $v_i^l \in V^l$ and edges $e_{ij}^l \in E^l$. In the model presented, we assume that all nodes have the capability to establish bypasses and each node $v_i^l \in V^l$ is connected to node $v_i^p \in V^p$ for simplicity. The topology information also contains information about the capacity of the links in the IP and the circuit layer, given by C_{ij}^l and C_{ij}^p respectively.

We also require the knowledge of IP routing as the bypassing of flows is constrained to ensure that the bypass endpoints must lie on the original routing path of the aggregate flows. Routing information can be obtained via passive monitoring solutions such as [12, 13] which are typically employed in IP networks. We use a routing model where the routing of an aggregate flow from router v_s^l to router v_d^l is defined by the parameter ψ_{ij}^{sd}, which indicates if the routing of the aggregate flow from v_s^l to v_d^l uses the link $e_{ij}^l \in E^l$. From this parameter, and the network topology information, we derive two extended parameters which are useful for bypass computation. They are:

- ψ_{xy}^{sd} : Boolean to indicate if the traffic from v_s^l to v_d^l uses the loose path $v_s^l \rightarrow v_x^l \rightarrow v_y^l \rightarrow v_d^l$. Note that $x \neq y$, $v_x^l, v_y^l \in V^l$.
- $\psi_{xy}^{sd}(ij)$: Boolean to indicate if the traffic from v_s^l to v_d^l uses the loose path $v_s^l \rightarrow v_x^l \rightarrow v_i^l \rightarrow v_j^l \rightarrow v_y^l \rightarrow v_d^l$, given that $e_{ij}^l \in E^l$, $x \neq y$, $v_x^l, v_y^l \in V^l$.

The first parameter ψ_{xy}^{sd} indicates if the route from v_s^l to v_d^l traverses over v_x^l and v_y^l, and therefore if a bypass from v_x^l to v_y^l can be used to re-route traffic from v_s^l to v_d^l. The parameter $\psi_{xy}^{sd}(ij)$ indicates if the link e_{ij}^l would be bypassed by a bypass from v_x^l to v_y^l for the flow from v_s^l to v_d^l. For example, in Fig. 2, the bypass from P to R is a valid bypass for the path from A to X, indicating that $\psi_{PR}^{AX} = 1$ and the bypass traverses link PQ indicating that $\psi_{PR}^{AX}(PQ) = 1$.

Bypass Establishment Service

In our model, we assume that all nodes in the IP layer have the capability to establish a bypass. This means that the nodes not only have the capability to establish a dynamic circuit in the underlying transport network, but also have the capability to re-route specific aggregate flows onto this circuit. Note that this circuit is not advertised in the IP layer, and hence routing rules at the ingress router must be configured accordingly. We also take into account the fact that the type of transport network used may constrain the granularities of the circuits established. While technologies such as carrier Ethernet and MPLS-TP can establish circuits with flexible granularities, legacy TDM services and

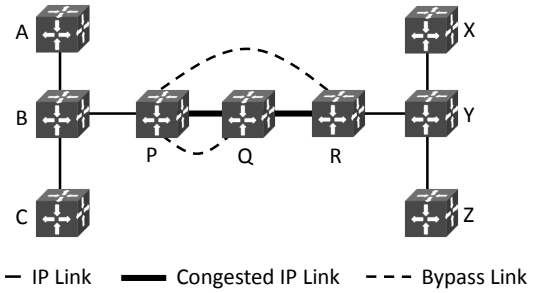

Figure 2: Example of routing constraints on Overlapping bypasses

pure WDM services may only support fixed granularities. In order to compute bypasses within these constraints, it is necessary to have information about the types of granularities supported by the transport network. For simplicity, in the model presented here, we assume that all nodes in the transport network can establish circuits with all granularities supported by this network. We assume that there are T different bypass granularities supported in the transport network, and for each granularity type $t \in T$, capacity of the bypass is given by C^t_{BP} and (interface) cost is given by $Cost^t$. The interface cost is a cost assigned to the endpoints of each bypass for establishment of the circuit, and while depending on the hybrid network architecture, a new interface may or may not be used to establish this circuit, we still assume that there is some cost involved with this operation. We also assign a cost per unit bandwidth P_{ij} for every link $e^p_{ij} \in E^p$ which in conjunction with the routing path used by the circuit, the capacity of the circuit and the interface cost determines the total cost of establishing the bypass.

Traffic Monitoring Service

The traffic monitoring service is an essential service for bypass computation. For the idealistic bypass computation model presented in this section, the traffic monitoring service must have the capability to provide accurate values of aggregate flows in the network. In the complete traffic matrix, traffic due to aggregate flows from a router v^l_s to v^l_d in the network is given by λ_{sd}. Note however that it is difficult to evaluate the actual traffic from all sources to all destinations in large networks. Therefore we also develop alternate optimization models which require limited traffic information which are presented in the next section.

Failure/Congestion Identification

As the proposed solution is targeted towards short-term congestion or failure events, the alarm system in place must have the capability to classify an event as a short-term or a long-term event. In congestion events, it is necessary to identify if the congestion is a result of long term traffic increase or due to short lived traffic surges. While the former is gradual and can be observed over time, the latter usually arrives suddenly (and possibly unexpectedly) and requires immediate action. For a failure event, backup mechanisms are typically installed in place to compensate for link or interface failures, but in events where these

backup mechanisms cannot be triggered, such as a router failure, the bypass computation can be invoked.

3.1. Bypass Computation with Complete Traffic Matrix

We now present an ILP based approach which computes the optimal bypass-based routing solution under the assumptions that the complete traffic matrix is known [2]. Three variables are used in this ILP which determine the position and granularity of a bypass in the IP layer and routing of the bypass circuit in the circuit network. They are:

- X^t_{xy} : Boolean to indicate if a bypass of type t exists from node v^l_x to v^l_y.
- f^{sd}_{xy} : Boolean to indicate if the aggregate flow from v^l_s to v^l_d is bypassed over a bypass from v^l_x to v^l_y.
- $r^t_{xy}(ij)$: Boolean to indicate if the bypass X^t_{xy} uses the link e^p_{ij} in the circuit layer.

In the ILP, we attempt to minimize the total cost of establishment of the bypasses. The ILP is modeled as :

$$Min : \sum_t Cost^t \sum_{xy} X^t_{xy} + \sum_t C^t_{BP} \sum_{ij} P_{ij} \sum_{xy} r^t_{xy}(ij) \tag{1}$$

Subject to Constraints:

$$\forall v^l_s, v^l_d, v^l_x, v^l_y \in V^l : f^{sd}_{xy} \leq \psi^{sd}_{xy} \tag{2}$$

$$\forall v^l_s, v^l_d \in V^l, e^l_{ij} \in E^l : \sum_{xy} \psi^{sd}_{xy}(ij) \cdot f^{sd}_{xy} \leq 1 \tag{3}$$

$$\forall v^l_s, v^l_d, v^l_x, v^l_y \in V^l : f^{sd}_{xy} \leq \sum_t X^t_{xy} \tag{4}$$

$$\forall v^l_x, v^l_y \in V^l : \sum_{sd} \lambda_{sd} \cdot f^{sd}_{xy} \leq \alpha \sum_t X^t_{xy} \cdot C^t_{BP} \tag{5}$$

$$\forall e^l_{ij} \in E^l : \sum_{sd} \lambda_{sd} \cdot \psi^{sd}_{ij} \left(1 - \sum_{xy} \psi^{sd}_{xy}(ij) \cdot f^{sd}_{xy}\right) \leq \alpha C^l_{ij} \tag{6}$$

$$\forall v^l_x, v^l_y \in V^l : \sum_t X^t_{xy} \leq 1 \tag{7}$$

$$\forall t \in T, v^p_x, v^p_y \in V^p, : \sum_i r^t_{xy}(xi) = X^t_{xy} \tag{8}$$

$$\forall t \in T, v^p_x, v^p_y \in V^p, : \sum_i r^t_{xy}(iy) = X^t_{xy} \tag{9}$$

$$\forall t \in T, v^p_x, v^p_y, v^p_i \in V^p, i \neq x, y : \sum_k r^t_{xy}(ki) = \sum_j r^t_{xy}(ij) \tag{10}$$

$$\forall e^p_{ij} \in E^p : \sum_t \left(C^t_{BP} \sum_{xy} r^t_{xy}(ij)\right) \leq C^p_{ij} \tag{11}$$

Eq. 2, 3 and 4 define the constraints for routing of flows over bypasses in the IP network. Eq. 2 indicates that traffic from v_s^l to v_d^l may be routed on a bypass from v_x^l to v_y^l if and only if the original route goes via nodes v_x^l and v_y^l. This constraint ensures that links not on the original path of an aggregate flow are not affected by the bypassing of the said flow. The constraint in Eq. 3 ensures that different bypasses chosen to reroute a given aggregate flow do not overlap. In case of an overlap of bypasses selected for the same aggregate flow, there will be at least one link e_{ij}^l which is bypassed by multiple bypasses for the same flow implying that $\sum_{xy} \psi_{xy}^{sd}(ij) \cdot f_{xy}^{sd} > 1$, and hence would violate the constraints. For instance, in Fig. 2 bypasses from P to R and from P to Q overlap each other and are a possible solution for the given problem. However, the flow from A to X cannot be bypassed over both bypasses simultaneously, and this solution will be eliminated as the constraint in Eq. 3 will be violated ($\psi_{PQ}^{AX}(PQ) \cdot f_{PQ}^{AX} + \psi_{PR}^{AX}(PQ) \cdot f_{PR}^{AX} = 2$). Finally, the constraint in Eq. 4 ensures that an aggregate flow can only be bypassed from v_x^l to v_y^l if a bypass exists between these nodes and Eq. 5 constraints the capacity used by flows rerouted on the bypass. Eq. 6 defines the link capacity constraint for the IP links in the network. The term $\lambda_{sd} \cdot \psi_{ij}^{sd}$ is used to determine if an aggregate flow from v_s^l to v_d^l uses the link e_{ij}^l while the term $\left[1 - \sum_{xy} \psi_{xy}^{sd}(ij) \cdot f_{xy}^{sd}\right]$ is used to determine if the specified flow is bypassed over the link. If the aggregate flow is bypassed, the sum $\sum_{xy} \psi_{xy}^{sd}(ij) \cdot f_{xy}^{sd} = 1$, as constrained in Eq. 3, and the traffic for that particular flow is not taken into consideration. Eq. 7 ensures that there is only one bypass established between a pair of nodes. This constraint is introduced primarily to reduce the complexity of the ILP, as if multiple bypasses were allowed between a pair of nodes, the aggregate flow assignment variable f_{xy}^{sd} would also be dependent on the class of bypass used to reroute the flow.

The constraints in Eq. 8, 9 and 10 define the routing of the bypasses in the circuit layer, with Eq. 8 and 9 ensuring that if a bypass of type t from node v_x^p to v_y^p exists, at least one outgoing link at the switch v_x^p and one incoming link at the switch v_y^p is used to route the bypass. Note here that v_x^l, v_y^l are connected to v_x^p and v_y^p respectively. Eq. 10 ensures the routing continuity of a bypass in the physical layer, and Eq. 11 constrains the capacity used on a physical link by different bypasses routed in the physical layer. The physical path variable $r_{xy}^t(ij)$ is also used in the objective function (Eq. 1) to compute the circuit bandwidth cost.

4. Adaptations for Incomplete Traffic Matrix Information

When modeling solutions for real IP networks, it is not realistic to assume complete knowledge of the traffic matrix in the network. However, alternate traffic measurements such as link loads are readily available in these networks. We now present ILP based solutions for the bypass-based routing problem when complete traffic matrix information is not known.

4.1. Bypass based computation using Measured Traffic Bounds

In this approach as presented in [3], we present an optimization model wherein we compute *upper bounds* on aggregate flows and use these bounds to compute bypasses in the network.

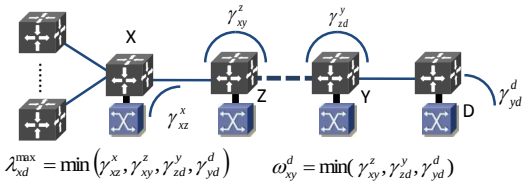

Figure 3: Traffic Measurement using virtual output queues

A primitive mechanism to measure upper bounds on traffic between a source-destination pair is to use the minimum link load along the routing path. Therefore, the upper bound λ^{max}_{sd} can be given by

$$\lambda^{max}_{sd} = min\left(LinkLoad(e^l_{ij}) : e^l_{ij} \in E^l, \psi^{sd}_{ij} = 1\right) \tag{12}$$

However, during the course of our investigation, we observed that these bounds were too prohibitive and were therefore not easily applicable for the bypass-based routing method. We therefore used an alternate traffic measurement which measures the traffic in the *virtual output queues* of the routers and found that this measurement can give us a more appropriate bound for λ^{max}_{sd}. We introduce a parameter γ^e_{df} as the traffic on the link e^l_{de} routed to link e^l_{ef} at node v^l_e, with special cases as the traffic inserted at a node v^l_e on an link e^l_{ef} indicated by γ^e_{ef} and the traffic destined for node v^l_e on an link e^l_{de} indicated by γ^e_{de}. These parameters can be measured by observing the virtual output queues of the routers, and can be used to estimate the max bound on aggregate traffic as

$$\lambda^{max}_{sd} = min\left\{\begin{array}{l}(\gamma^y_{xz} : y \neq s, d, \psi^{sd}_{xy} = \psi^{sd}_{yz} = 1)\\(\gamma^s_{sx} : \psi^{sd}_{sx} = 1)\\(\gamma^d_{yd} : \psi^{sd}_{yd} = 1)\end{array}\right\} \tag{13}$$

To illustrate this measurement, take the example presented in Fig. 3: The traffic from v^l_X to v^l_D will be less than the traffic inserted at v^l_X onto the link e^l_{XZ} (γ^X_{XZ}), the traffic forwarded from the link e^l_{XZ} to e^l_{ZY} (γ^Z_{XY}) and from link e^l_{ZY} to e^l_{YD} (γ^Y_{ZD}) and finally the traffic leaving the network at node v^l_D coming from the link e^l_{YD} (γ^D_{YD}).

It is also interesting to note in Fig 3 that if the routing in the network was Open Shortest Path First (OSPF), with an assumption that equal cost multi-path routing was not used, and if the term γ^s_{sx}, (the traffic injected at

the source), was not taken into consideration, the expression in Eq. 13 would provide a bound for all traffic upstream and from the node v_s^l to the destination v_d^l, as the routing path of all upstream traffic flows passing from v_s^l to v_d^l is same as the routing path from v_s^l to v_d^l. As is typically the case in current IP networks, routing decisions are based purely on destination IP addresses and typically follow a single path mechanism. Therefore, as illustrated in Fig. 3, a new parameter ω_{pq}^d is introduced which computes the total flow to destination v_d^l that can be bypassed v_p^l to v_q^l and is given by:

$$\omega_{pq}^d = \psi_{pq}^{pd} \cdot min \left\{ \begin{array}{l} (\gamma_{xz}^y : y \neq p, d, \psi_{xy}^{pd} = \psi_{yz}^{pd} = 1) \\ (\gamma_{yd}^d : \psi_{yd}^{pd} = 1) \end{array} \right\} \tag{14}$$

As shown in the example in Fig. 3, for all aggregate flows from sources upstream and at X to D, if the max bound on traffic is determined by the traffic forwarded between consecutive links between X and D, the max bound for individual flows would be the same as the max bound for aggregate flows. Therefore, it is easy to recognize that the max bound for aggregate flows is at least equal, but likely tighter than the sum of individual max bounds in a very large number of cases. We can therefore modify our bypass mechanism to take into consideration bypassing of *all aggregate flows* from and upstream from the bypass ingress site to a given destination at or downstream from the bypass egress site. It should also be noted that in the absence of measurements from the virtual output queues, the upper bound on all aggregate flows may also be computed as

$$\omega_{pq}^d = \psi_{pq}^{pd} \cdot min \left(LinkLoad(e_{ij}^l) : e_{ij}^l \in E^l, \psi_{ij}^{pd} = 1 \right) \tag{15}$$

4.1.1. Bypass Computation using Bounds on Aggregate Upstream Traffic

As mentioned above, in this approach all aggregate flows from and upstream of the bypass ingress to a destination are rerouted onto a bypass. The computation of bounds for all aggregate flows is far tighter than the bounds on individual flows, as can be seen from Eq. 13 and Eq. 14. We use these bounds to estimate the worst case traffic on links and bypasses, however as we do not consider unique aggregate flows in our computation, some additional parameters have to be taken into consideration, which are described below:

Residual Flows after Bypass: As described above, we now bypass all traffic to a destination from the ingress of the bypass. Now consider the scenario as shown in Fig. 4: a bypass is established from router X to Y towards destination D. While the bypass reroutes all aggregate flows from and upstream of X to D across the link $Z - Y$, the link $Z - Y$ still has aggregate flows for D which are inserted at and upstream from X‘ and at Z for D. In order to compute the traffic on a link after flows upstream of the link have been bypassed, we introduce a new parameter $\hat{i}_{xy}^d(ij)$ which estimates the *maximum residual traffic bound* on the link e_{ij}^l to the destination v_d^l, if the bypass from v_x^l to v_y^l is used to reroute all traffic to v_d^l.

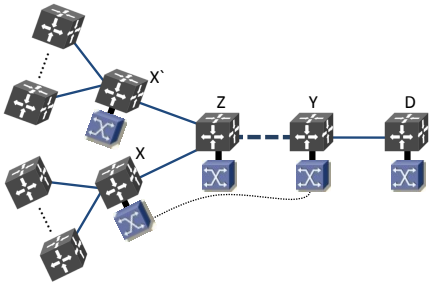

Figure 4: Residual Traffic on Congested link after bypassing all aggregate flows from and upstream of bypass ingress

Algorithm 1: Algorithm to compute $\hat{i}^d_{xy}(ij)$

```
if (ψ^{xd}_{xy}(ij) == 0) then
    î^d_{xy}(ij) = 0
else
    if (x == i) then
        î^d_{xy}(ij) = 0
    else
        î^d_{xy}(ij) = 0
        for (p s.t. ψ^{xd}_{xi}(pq) == 1 or ψ^{xd}_{xi}(qp) == 1 for some q, p ≠ x) do
            for (r s.t. e^l_{rp} ∈ E^l, ψ^{xd}_{xi}(rp) == 0) do
                î^d_{xy}(ij) += ω^d_{rp}
            end
            î^d_{xy}(ij) += λ^{max}_{pd} · ψ^{pd}_{pq}
        end
        if (î^d_{xy}(ij) > ω^d_{ij}) then
            î^d_{xy}(ij) = ω^d_{ij}
        end
    end
end
```

The parameter $\hat{i}^d_{xy}(ij)$ can be computed as the traffic inserted by all routers downstream from v^l_x till v^l_i to v^l_d as well as the traffic for flows which have one or more of the nodes downstream from v^l_x till v^l_i but not the node v^l_x in their routing paths. For example, in Fig. 4, the max bound for traffic to D traversing the link $Z-Y$ can be computed as $\omega^D_{X`Z}$ (traffic from and upstream of $X$`) $+\lambda^{max}_{ZD}$ (aggregate flow traffic inserted at the node Z for destination D). We use Algo. 1 to compute $\hat{i}^d_{xy}(ij)$. The algorithm first checks if rerouting of aggregate flows to destination v^l_d over the bypass from v^l_x to v^l_y affects any flows on the link e^l_{ij}, and if not, the residual flow $\hat{i}^d_{xy}(ij)$ is set to 0. Also, if the bypass originates at the ingress node v^l_i $(x == i)$, all traffic to destination v^l_d is bypassed and therefore traffic to v^l_d on link e^l_{ij} is equal to 0. For the scenario shown in Fig. 4, the algorithm traverses all nodes along the path from v^l_x to v^l_z (v^l_x not included),

and includes the max bounds for the traffic inserted at these nodes (λ_{zd}^{max}) and the traffic from all neighboring nodes connected to these nodes. Note here that the sum of max bounds is limited by the max bound of total traffic on link $Z-Y$ towards D given by ω_{ZY}^{d}.

In the new ILP formulation, as all flows to a particular destination are bypassed now at the bypass ingress, the flow indicator variable f_{xy}^{sd} is replaced by f_{xy}^{d}, which indicates if the flow to destination v_d^l is bypassed over the bypass from v_x^l to v_y^l. The objective function and the physical routing constraints remain unchanged in this formulation as they only depend on the variable X_{xy}^{t} which determines if a bypass is required between the nodes v_x^p and v_y^p.

$$Min : \sum_{t} Cost^{t} \sum_{xy} X_{xy}^{t} + \sum_{t} C_{BP}^{t} \sum_{ij} P_{ij} \sum_{xy} r_{xy}^{t}(ij) \tag{16}$$

Subject to Constraints:

$$\forall v_d^l, v_x^l, v_y^l \in V^l : f_{xy}^{d} \leq \psi_{xy}^{xd} \tag{17}$$

$$\forall v_d^l \in V^l, e_{ij}^l \in E^l : \sum_{xy} \psi_{xy}^{xd}(ij) \cdot f_{xy}^{d} \leq 1 \tag{18}$$

$$\forall v_d^l, v_x^l, v_y^l \in V^l : f_{xy}^{d} \leq \sum_{t} X_{xy}^{t} \tag{19}$$

$$\forall v_x^l, v_y^l \in V^l : \sum_{d} \omega_{xy}^{d} \cdot f_{xy}^{d} \leq \alpha \sum_{t} X_{xy}^{t} \cdot C_{BP}^{t} \tag{20}$$

$$\forall e_{ij}^l \in E^l : \sum_{d} \omega_{ij}^{d} \cdot \left(1 - \sum_{xy} \psi_{xy}^{xd}(ij) \cdot f_{xy}^{d}\right) + \sum_{d} \sum_{xy} f_{xy}^{d} \cdot \hat{\imath}_{xy}^{d}(ij) \leq \alpha C_{ij}^{l} \tag{21}$$

$$\forall v_x^l, v_y^l \in V^l : \sum_{t} X_{xy}^{t} \leq 1 \tag{22}$$

$$\forall t \in T, v_x^p, v_y^p \in V^p, : \sum_{i} r_{xy}^{t}(xi) = X_{xy}^{t} \tag{23}$$

$$\forall t \in T, v_x^p, v_y^p \in V^p, : \sum_{i} r_{xy}^{t}(iy) = X_{xy}^{t} \tag{24}$$

$$\forall t \in T, v_x^p, v_y^p, v_i^p \in V^p, i \neq x, y : \sum_{k} r_{xy}^{t}(ki) = \sum_{j} r_{xy}^{t}(ij) \tag{25}$$

$$\forall e_{ij}^p \in E^p : \sum_{t} \left(C_{BP}^{t} \sum_{xy} r_{xy}^{t}(ij)\right) \leq C_{ij}^{p} \tag{26}$$

The routing constraints in the IP layer are presented in Eq. 17, 18 and 19. Eq. 17 ensures that the router v_y^l is on the original routing path from v_x^l to v_d^l. Eq. 18 ensures that traffic to v_d^l is not bypassed by overlapping bypasses. Unlike the previous ILP, overlapping bypasses do not lead to routing misconfigurations here. For instance, in Fig. 4, a bypass from X to Y and a bypass from Z to Y can both be used to bypass traffic to the destination D. However, given that we only have maximum bound information for traffic bypassed at X and

Z, the estimate of residual traffic is very lax leading to inefficient solutions. The third routing constraint (Eq. 19) ensures that aggregate flows can only be bypassed from v_x^l to v_y^l if a bypass exists between the routers. The bypass capacity constraint in Eq. 20 uses the max. bound of traffic from and upstream of the ingress node v_x^l to the destination v_d^l (ω_{xy}^d) to evaluate the required bypass capacity. In the link capacity constraint (Eq. 21) traffic on a link is expressed as a sum of max bounds of traffic to all destinations: If there is no bypass, $\left(\left(1 - \sum_{xy} \psi_{xy}^{xd}(ij) \cdot f_{xy}^d\right) = 1\right)$, the max. bound of all traffic to destination v_d^l i.e. (ω_{ij}^d) is used. If there is a bypass from v_x^l to v_y^l, the max. bound on the residual traffic to the destination v_d^l i.e. $\hat{\imath}_{xy}^d(ij)$ is used. As max. traffic bounds are used, estimated traffic on a link may be more than the actual traffic, and hence in our implementation, this constraint is only applied to congested links which are known apriori, as load on non-congested links cannot be increased by bypass establishment. It is very important to note here that the requirement of ensuring that the egress of the bypasses lies on the original routing path ensures that the link load of existing links can only decrease, and therefore enables us to establish bypasses even with poor upper bound estimates (albeit at low bypass capacity utilization). If however, these bounds were to be used for other routing techniques, poor upper bounds would significantly degrade the performance of these architectures as link capacity constraints using max bounds would have to be computed on *all links* in the network.

4.2. Bypass Based Computation using Estimated Traffic Matrices

By using the traffic bounds as shown in Section 4.1, we can compute bypasses and the appropriate flows to be re-routed. However, it was observed that the use of bounds does lead to very high bypass capacities as compared to the ideal bypass based routing model presented in Section 3.1. We therefore used traffic estimation techniques to obtain an approximate traffic matrix using known traffic measurements, and used these measurements to compute the bypass based routing. In this paper, we experiment with and use the *Tomogravity model* presented in [4] and modify the same to use traffic measurements from virtual output queues, which provides tighter constraints, instead of measurements from link loads as used in the previous model. To use this model, we first estimate the traffic matrix using the gravity model (λ_{sd}^G), i.e., :

$$\forall v_s^l, v_d^l \in V^l, s \neq d : \lambda_{sd}^G = \sum_{y:e_{sy}^l \in E^l} \gamma_{sy}^s \cdot \frac{\sum_{y:e_{yd}^l \in E^l} \gamma_{yd}^d}{\sum_{(p:v_p^l \in V^l, p \neq s)} \sum_{(y:e_{yp}^l \in E^l)} \gamma_{yp}^p} \tag{27}$$

We now define a quadratic optimization problem based on the tomogravity model using the weighted least squares method [4] to estimate the traffic matrix (λ_{sd}^{TG}). We define the tomogravity traffic estimate as

$$\forall v_s^l, v_d^l \in V^l, s \neq d : \lambda_{sd}^{TG} = \lambda_{sd}^G \cdot (1 + G_{sd}) \tag{28}$$

The aforementioned notation is used so as to reduce the quadratic optimization objective function to

$$Min : \sum_{s,d:v_s^l,v_d^l \in V^l, s \neq d} G_{sd}^2 \tag{29}$$

using the following constraints

$$\forall v_s^l, v_d^l \in V^l, s \neq d : G_{sd} \geq -1 \tag{30}$$

$$\forall e_{pq}^l, e_{qr}^l \in E^l : \sum_{s,d:\psi_{pq}^{sd}=\psi_{qr}^{sd}=1} \lambda_{sd}^G \cdot (1 + G_{sd}) = \gamma_{pr}^q \tag{31}$$

$$\forall e_{sp}^l \in E^l : \sum_{d:\psi_{sp}^{sd}=1} \lambda_{sd}^G \cdot (1 + G_{sd}) = \gamma_{sp}^s \tag{32}$$

$$\forall e_{pd}^l \in E^l : \sum_{s:\psi_{pd}^{sd}=1} \lambda_{sd}^G \cdot (1 + G_{sd}) = \gamma_{pd}^d \tag{33}$$

The objective function in Eq. 29 minimizes the weighted least square distance from the gravity model estimate. The constraint in Eq. 30 ensures that the traffic on every link is positive, and the constraints in Eqns. 31, 32 and 33 define the equality constraints for the measurements obtained from the virtual output queues. The constraint in Eq. 31 ensures that sum of the estimated aggregate flows using both links e_{pq}^l and e_{qr}^l is equal to the measurement made at node v_q^l for traffic from node v_p^l and forwarded to v_r^l (γ_{pr}^q), and the constraints in Eq. 32 and 33 representing similar constraints on the aggregate flow estimates for traffic entering the network at the node v_s^l onto link e_{sp}^l and exiting the network at node v_d^l via link e_{pd}^l respectively.

Using the traffic matrix estimated with the tomogravity model, we can now compute bypasses with the optimization models presented in Sections 3.1 and 4.1.1. For application in the ideal bypass computation model presented in Section 3.1, the traffic λ_{sd} can be substituted by estimated traffic λ_{sd}^{TG}. However, in order to account for the errors in measurement, it is advisable to use a smaller congestion threshold α in both the bypass and link capacity constraints (Eq. 5 and 6 respectively).

In order to use the estimated traffic matrix in the optimization model presented in Section 4.1.1, it is necessary to compute values for the parameters ω_{pq}^d and $\hat{i}_{xy}^d(ij)$. These parameters are now computed as:

$$\omega_{pq}^d = \psi_{pq}^{pd} \cdot \sum_{s:\psi_{pd}^{sd}=1} \lambda_{sd}^{TG} \tag{34}$$

$$\hat{i}_{xy}^d(ij) = \psi_{xy}^{sd}(ij) \cdot \sum_{s:\psi_{ij}^{sd}=1,\psi_{xy}^{sd}=0} \lambda_{sd}^{TG} \tag{35}$$

The use of the estimated traffic matrix provides significantly lower values for the parameters ω_{pq}^d and $\hat{i}_{xy}^d(ij)$. However, as these values are prone to

errors (including underestimation of traffic), we must reduce the link utilization threshold parameter α used in the link and bypass capacity constraints in Eqns. 21 and 20 respectively.

5. Numerical Study

In this section we present a numerical analysis to compare the performance of the bypass based routing models presented. We use the Atlanta Reference network[14], shown in Fig. 5 as the IP network topology, and the transport network topology is also assumed to be the same. We assume a base traffic matrix as shown in Table 1, and assume that OSPF routing is used in the network. Using these values, we then determine the link capacities so that initial link utilization of all links is 70%. In order to induce congestion in the network, we randomly select a number of unique source-destination pairs and increase the traffic on these pairs by 150%. In the transport network, all links have a total available capacity of 100,000 units and a normalized cost per unit bandwidth $P_{ij} = 1$. We use nine different bypass types with capacity and normalized interface costs as shown in Table 2.

We solve the four bypass based routing models, henceforth termed as *bypass-based routing of individual (s,d) flows (BBR-I)*, *bypass based routing with traffic bounds (BBR-TB)*, *bypass-based routing of individual (s,d) flows with traffic estimates (Estimated-I)* and *bypass based routing with traffic bounds computed using traffic estimates (Estimated-TB)*, using the GNU Linear Programming Kit [15]. To solve the quadratic optimization problem to estimate the Tomogravity estimates, we use MATLAB's Optimization Toolbox [16].

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	5981	816	563	431	1592	444	576	491	563	252	695	324	1556	695
2	7132	0	2644	2239	755	6102	1832	4930	253	1952	755	1090	563	419	1855
3	977	3151	0	312	240	252	38	1149	85	97	97	145	85	37	431
4	678	2675	370	0	982	922	72	158	72	240	145	145	1521	157	192
5	512	905	286	1177	0	98	37	120	61	61	37	86	85	38	86
6	1903	7275	298	1106	121	0	97	874	86	575	181	132	158	240	264
7	536	2189	49	84	48	120	0	61	265	193	97	60	61	73	192
8	691	5884	1368	192	143	1047	73	0	204	181	97	145	157	241	516
9	584	299	108	84	73	108	322	250	0	204	73	276	61	120	252
10	678	2331	120	286	73	690	228	215	250	0	419	1209	910	1137	670
11	298	905	120	180	49	215	120	120	84	500	0	240	612	384	157
12	833	1298	180	179	108	156	72	179	334	1439	286	0	97	97	934
13	393	678	107	1819	107	192	72	191	72	1082	727	121	0	313	193
14	1855	500	48	191	49	286	85	287	143	1355	464	120	370	0	97
15	833	2212	512	227	108	321	227	619	299	797	191	1118	228	120	0

Table 1: Initial traffic matrix in the Atlanta Network

Capacity	If. Cost	Capacity	If. Cost	Capacity	If. Cost
100	200	1,000	500	10,000	800
200	300	2,000	600	20,000	900
500	400	5,000	700	50,000	1000

Table 2: Different bypass types with corresponding capacities and interface costs

In this analysis, we compare various parameters such as average number of bypasses established, capacity installed and number of flows bypassed by the different mechanisms in the network. In order to generate congestion in the

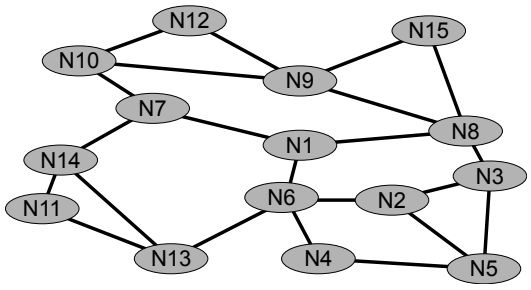

Figure 5: The Atlanta Reference Network [14]

network, we vary the total number of flows which observe traffic increase and for each scenario the average is computed over 20 iterations. We measured the error between the tomogravity estimates and the actual traffic for the aggregate flows over a large number of matrices, and saw that the mean standard deviation for the abovementioned difference in the given network was about 10%. In order to compensate for this error, we used lower values of the congestion threshold α for the link and the bypass matrices when using the tomogravity model based methods (Estimate-I and Estimate-TB).

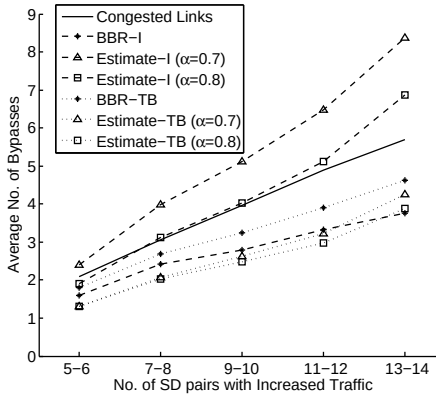

Figure 6: Number of bypasses for different no. of flows with increased traffic

Fig. 6 shows the total number of bypasses required by the different methods. We observe that the total number of bypasses required when the exact traffic matrix is known (*BBR-I*) is lower than the total number of congested links in the network, indicating that the proposed solution is better than basic link capacity increase methods. This indicates that atleast on bypass established in this solution bypasses two or more congested links in the network. However, reduction in the congestion thresholds ($\alpha = 0.7, 0.8$) when using the *Estimate-I* method leads to significant increase in the number of bypasses required, and in most cases, the number of bypasses required are higher than the number of

congested links. On the other hand, when comparing the *BBR-TB* model with its counterpart *Estimate-TB*, it is seen that the number of bypasses established in the *Estimate-TB* mechanism are lower than the *BBR-TB* method. This is observed as the traffic bounds on the parameters $\hat{i}^d_{xy}(ij)$ and ω^d_{pq} (which are the measures of the residual and aggregate downstream traffic in the model) are likely to be lower when using the tomogravity model (Eq. 34 and 35) as compared to using the measurements from the Virtual Output Queues (Algo. 1 and Eq. 15). Therefore, with even lower thresholds on α, we observe that the total number of bypasses are significantly lower.

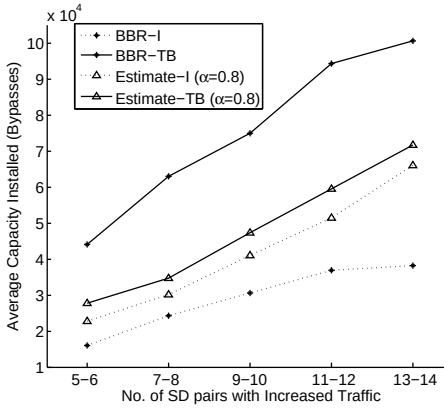

Figure 7: Average capacity of the installed bypasses for different no. of flows with increased traffic

On further observation, we see that the total number of bypasses installed by the *Estimate-TB* methods are also lower than the number of bypasses installed in the *BBR-I* method. However, the *Estimate-TB* method, on average, bypasses more flows than the *BBR-I* method, as it must bypass all flows from and upstream from the bypass ingress to the bypass destination. As a consequence, it is observed that the *Estimate-TB* method tends to use *fewer bypasses with larger capacities* as compared to the *BBR-I* method. This phenomenon is shown in Fig. 7, where we see that while the total number of bypasses may be higher, the total capacity of installed bypasses by the *BBR-I* method is the lowest. We also observe that the *Estimate-TB*and the *Estimate-I* method with congestion thresholds at ($\alpha = 0.8$) perform better than the *BBR-TB* method, which is again due to the use of very conservative values used when computing the bounds on traffic $\hat{i}^d_{xy}(ij)$ and ω^d_{pq}. The *Estimate-I* method performs marginally better than the *Estimate-TB* method, as it can bypass flows based on the source as well as the destination, while the *Estimate-TB* method is constrained on bypassing all flows to a given destination from the bypass ingress. However, the difference in the two models is marginal indicating that the use of more accurate bounds in the *BBR-TB* method can significantly reduce the capacity required for this method. At lower values of the congestion thresh-

old ($\alpha \leq 0.7$), the Estimation based models perform worse than the *BBR-TB* models. This indicates that the Estimation based methods are not applicable when the error in the traffic matrix estimates is high, as this would require lower congestion thresholds.

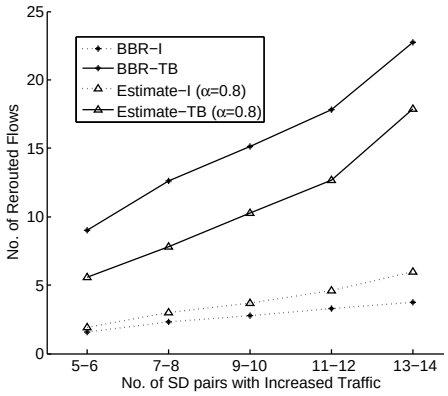

Figure 8: Number of aggregate flows rerouted for different no. of flows with increased traffic

One of the major advantages of the *BBR-TB* and the *Estimate-TB* based methods is that the implementation of this model does not require special switching capabilities at the bypass ingress routers, as all routers can switch packets based of destination IP address masks, but special switching rules/functions are required when switching packets based on source and destination IP address masks at the bypass ingress. On the other side however, we see that the *BBR-TB* and the *Estimate-TB* based methods lead to higher number of flows being bypassed. As seen in Fig. 8, the *BBR-I* and the *Estimate-I* models lead to very small number of flows being bypassed. In fact, the number of bypassed flows is lower than the number of flows experiencing congestion in the network. However, the *BBR-TB* and the *Estimate-TB* models lead to a larger number of flows being bypassed. However, it was observed that in the *Estimate-TB* and the *BBR-TB* models, the number of destinations selected at the bypasses (f^d_{xy}) were comparable to the total number of flows bypassed in the the *BBR-I* and the *Estimate-I* models. Therefore, this model is appropriate for networks where the monitoring functions are governed only by the egress router for the traffic. In such a scenario, the number of destinations for which any flow was bypassed would be the critical factor rather than the total number of flows that were bypassed.

As mentioned before, since the error observed in the measured traffic and the actual traffic was significant (about 10%), in some cases we observed that the link or bypass capacities were underestimated in the the *Estimate-TB* and the *Estimate-I* models. We observed cases in which bypass and link utilization values when using the correct traffic matrix was found to be higher than the maximum congestion threshold ($\alpha = 0.9$). In Fig. 9 we indicate the fraction of

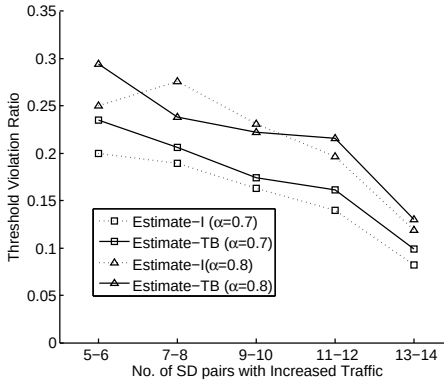

Figure 9: Ratio of threshold violations observed for different no. of flows with increased traffic

congested links or bypasses that were found to violate the maximum congestion threshold after actual traffic values were applied to the solutions computed using the *Estimate-TB* and the *Estimate-I*.

It was observed that the threshold violations decreased with increase in the number of source-destination pairs experiencing increased traffic. This phenomenon was observed primarily due to the fact that with the increase in number of overloaded source-destination pairs, more flows were bypassed and simultaneously, higher capacity bypasses were used. The higher capacity of bypasses used ensured that the bypass was tolerant to higher absolute errors, and the nature of the bypass granularity also indicated that higher capacity bypasses typically experienced lower utilization, therefore being more tolerant to errors in estimation. On the other hand, significant number of threshold violations were observed especially in the cases when small capacity bypasses were used. In order to curb this problem, we can employ the use of a variable value of the congestion threshold, with a lower congestion threshold α at low bypass and link capacities to provide a larger margin for absolute errors in traffic estimation, while higher values of the congestion threshold can be used for bypasses and links with high capacities.

Our results show that the bypass based routing mechanism can be used to counter short term congestion events in the network. The *BBR-I*, *BBR-TB* and the *Estimate-TB* mechanisms require fewer number of bypasses than the number of congested links, indicating that in these solutions, atleast one bypass crosses multiple congested links in the IP network, and therefore is better than the traditional mechanisms used which only increase link capacity. Note that the bypass based mechanisms can also be modeled to counter congestion at nodes or node failures, while still maintaining routing stability in the rest of the network. The estimation based method *Estimate-TB* show better performance than the *BBR-TB* due to the reduced traffic thresholds, and show that better estimation of the traffic bounds can significantly improve the performance of the *BBR-TB* method.

On the other hand, the lower congestion threshold used in the *Estimate-I* method leads to increase in the number of bypasses and average bypass capacity when compared with the ideal *BBR-I* mechanisms. However, the difference in the two models was marginal, and if better estimation mechanisms were to be employed, the *Estimate-I* method can perform almost as well as the *BBR-I* method.

6. Conclusion

We presented a new network engineering paradigm which uses what we refer to as *optical bypasses* in the circuit-switched network to alleviate congestion in the IP layer while keeping the IP routing stable. Our results show that the number of bypassed flows is relatively small, indicating that network operators are not subject to heavy reconfigurations during bypass establishment and teardown. The results show that even without the knowledge of the traffic matrix, the proposed ILP can compute bypasses, albeit at a higher cost due to the typically higher capacity of the resulting bypasses. We also see that use of traffic estimates, such as tomogravity model, can significantly improve the performance of the bypass based routing solution. As the next step, it is necessary to develop more efficient traffic estimation mechanisms as well as mechanisms to predict short term congestion events.

References

[1] Internet2 Headroom Practice: `https://wiki.Internet2.edu/confluence/down-load/attachments/17383/Internet2+Headroom+Practice+8-14-08.pdf`

[2] M. Chamania, M. Caria, A. Jukan, "*Achieving IP Routing Stability with Optical Bypass,*" IEEE ANTS 2009.

[3] M. Chamania, M. Caria, A. Jukan, "*Effective Usage of Dynamic Circuits for IP Routing,*" IEEE ICC 2010.

[4] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "*Fast Accurate Computation of Large-scale IP Traffic Matrices from Link Loads,*" ACM SIGMETRICS 2003.

[5] The Lambda Station Project: `http://www.lambdastation.org/`

[6] Policy Based Routing - Cisco White Paper, `http://www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a4409.shtml`

[7] G. Myoung Lee, et al., "*Performance Evaluation of an Optical Hybrid Switching System,*" IEEE GLOBECOM 2003.

[8] C. Xin, C. Qiao, Y. Ye, S. Dixit, "*A Hybrid Optical Switching Approach,*" IEEE GLOBECOM 2003.

[9] I. de Miguel, et al., "*Polymorphic Architectures for Optical Networks and their Seamless Evolution towards Next Generation Networks,*" Photonic Network Communications, Volume 8, Number 2, September 2004

[10] OpenFlow Project, `http://www.openflowswitch.org/`

[11] L. Cheng, J. Ellson, A. Jukan, P. Lamy, E. Varma, "*Network Engineering-Control of Dynamic Link Topology in User Networks,*" Bell Labs Technical Journal (BLTJ), Vol. 8, No 1, pp. 207-218, July 2004.

[12] A. Shaikh and A. Greenberg, "*OSPF Monitoring: Architecture, Design and Deployment Experience,*" 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1 (NSDI'04), Vol. 1.

[13] D. Watson, F. Jahanian, and C. Labovitz, "*Experiences With Monitoring OSPF on a Regional Service Provider Network,*" IEEE ICDCS 2003.

[14] Atlanta reference Network, `http://sndlib.zib.de/coredata.download.action?objectName=atlanta\&format=native\&objectType=network`

[15] GLPK (GNU Linear Programming Kit), `http://www.gnu.org/software/glpk/`

[16] MATLAB Optimization Toolbox Documentation, `http://www.mathworks.com/access/helpdesk/help/toolbox/optim/`