A Novel Approach to Accurately Compute an IP Traffic Matrix Using Optical Bypass

Marcel Caria and Admela Jukan

Technische Universität Carolo-Wilhelmina zu Braunschweig

Email: {caria, jukan}@ida.ing.tu-bs.de

Abstract—The traffic matrix is not readily available in IP networks and it can neither be calculated from link loads and routing information due to the ill-posed nature of the correspondent linear system, nor can it be estimated accurately enough. In this paper, we propose a novel approach to measure individual origin-destination flows using optical bypass to increase the rank of the said linear system, which let us accurately calculate a large portion of the IP traffic matrix. The method is inspired by optical bypass techniques known from multi-layer traffic engineering in IP-optical systems, which we use in this novel context. The traffic matrix calculation is possible due to the manner in which we create bypasses, which makes our method effective and unique. Our results show that already a small number of bypasses is sufficient to obtain almost the entire traffic matrix, while the impact of our method on network design and operation is negligible.

I. INTRODUCTION

The knowledge of the traffic matrix is essential for IP network operation, including functions of traffic engineering, routing protocol configuration, and fault diagnosis. At the same time, it is a well known fact that the traffic matrix, i.e., defined by the traffic demand of all origin-destination router pairs (or shorter: OD flows), is not readily available in IP networks. While the traffic flows between any pair of routers can be measured, these measurements are not practical for large networks, as they would require a significant amount of monitoring equipment and network-wide configuration efforts [1]. Therefore, the traffic matrix is usually estimated based on link load measurements and the routing information.

To calculate the traffic matrix based on link load measurements and the routing, a solution needs to be found of the under-determined (i.e., ill-posed) linear inverse problem $L = R \cdot F$, where the link load of n links in the network is defined as column vector $L = (l_1, l_2, \ldots, l_n)^T$, the demand of all m OD flows is a row vector $F = (f_1, f_2, \ldots, f_m)$, while the routing information is a binary $n \times m$ matrix R, with r_{ij} is 1 if flow j is routed via link i, and 0 otherwise. Usually, link loads are measured with the SNMP protocol, while the routing information is known. The above linear system is always under-determined. For instance, in the pan-European network topology with 37 nodes and 57 links [2], one could measure 114 link loads, but there are 1332 OD flows, resulting in a very low rank of the linear system, i.e., with a high number of columns and a low number of rows.

In this paper, we propose a novel approach to measure individual OD flows using IP link bypass with optical circuits. The method is inspired by optical bypass techniques know from multi-layer traffic engineering in IP-optical systems, where an optical bypass (circuit) can be created between routers to boost the network capacity [3]. Here, we use the bypass not for the purpose of traffic engineering, but to reroute selected flows, thus obtaining their measured loads via SNMP. By measuring the flows on the bypass we can reduce the under-determination of the linear system $L = R \cdot F$, which in turn allows for a more accurate traffic matrix estimation. As we will show, only a few optical bypasses need to be setup to obtain a high number of OD flows, such as 7 in a 37 nodes network, where we are able to obtain the accurate traffic matrix for more than 70% of all OD flows. The bounds tightening algorithm, which is part of our method, allows us for the OD flows that remain unsolved to at least determine upper and lower bounds which can be used as constraints for successional traffic matrix estimation. It is precisely the mechanisms of optical bypass setup that make the idea highly suitable for measurements, which is not only fundamentally different from any previous application on IP-optical networking, but also with respect to the existing measurement-based methods.

The rest of the paper is organized as follows: Section II discusses the related work and our contribution. Section III presents the IP-optical network architecture, and shows per-flow measurements on a bypass. We present an analytical model for bypass location optimization in Section IV. Section V presents the performance study and Section VI concludes the paper.

II. RELATED WORK AND OUR CONTRIBUTION

Multiple traffic matrix estimation techniques seek the solution of the linear system $L = R \cdot F$, a group of techniques generally referred to as network tomography. Since the linear system is ill-posed, it has multiple solutions, so the accuracy of network tomography methods differ based on the assumptions. One typical approach is to assume a certain traffic distribution function. Another method is to derive a solution with higher order statistics of the link loads, linear programming, or quadratic programming [4]. Yet another known technique is the *gravity model* [5]. Here, the traffic matrix is derived only from the total traffic entering the network at each ingress and the total traffic exiting the network at each egress, whereas the

interior network links and routing information are not considered. The gravity model can be used as input to the tomography method, also coined as *tomogravity model* [6]. Note that, regardless of the method, all proposed traffic matrix estimations typically exhibit average errors in the range of 10% to 25% with some OD flow estimate errors above 100% [7].

To increase the rank of the linear system, it is also possible to reconfigure the routing in the network. The authors in [7] propose to change the routing by altering the link loads in order to create an additional linear system $R \cdot \lambda = L$ (containing different R and L). This new linear system can be combined with the original one, which increases the rank. This method can be performed repeatedly until the desired rank is achieved. The authors in [8] propose to route flows over fixed network monitor nodes.

OD flows can also be directly measured. For instance, MPLS would use LSPs tunnels, OpenFlow could be configured to route on a per-OD-flow basis, in PBB-TE we can provision an E-Line, and Ethernet can use VLANs. Let's consider the case of MPLS first; here, the operator can set up an LSP between a pair of routers and install a packet counter on that LSP. In a network with N nodes, this would require the setup of $N \cdot (N-1)$ LSPs, and each LSP setup would involve all routers along the routing paths. In addition to this significant configuration overhead, which would be rather impractical also, it should be noted that MPLS – despite its maturity – is used in only 7% of all autonomous systems in the Internet and the Internet backbone is still primarily based on pure IP [9]. Considering E-Lines and VLANs, the same argument regarding the configuration overhead is true.

In the case OpenFlow routers are deployed in the IP network, it is indeed possible to measure an OD flow directly on the ingress router. The network operator could add a flow table entry for every egress router at the ingress router for this monitoring purpose only. However, this would require that the OpenFlow implementation in the router supports subnet masks on the IP destination field of flow table entries, which is not generally the case [10]. More importantly, the OpenFlow router – or software defined networking in general – is still in its infancy, and its deployment in IP backbones is not expected in the near future.

Another way to measure OD flows directly provides Cisco's IOS NetFlow feature, but the impact on the CPU load in the router can be significant [11], and NetFlow must be available on the routers. This is not generally the case, especially in the carrier network where multiple router vendor's equipment is used; it is well known that not every router vendor supports NetFlow. In addition, since network operators usually reduce NetFlow's CPU load by sampling (e.g., measuring only one out of 1000 packets), it has to be considered that the derived data on OD flows pose estimates only, whereas our approach yields real measurements.

The most straightforward approach to measure the entire traffic matrix is to monitor all ingress flows on all border routers. A cost-effective and simple solution is to mirror a border router's ingress traffic to the monitoring port, so that it can be analyzed and pre-processed by a low-end host connected to that monitoring port. A central server can then collect the data from all monitoring hosts, and the routing information can be taken from all routers (using SNMP) to perform network optimizations (e.g., traffic engineering). The issue with this solution is that it can not scale to current transmission speeds in core networks, with typically 100 Gbit/s, whereas the current network adapter cards support speeds up to 10 Gbit/s. However, there are systems on the market that do scale to core network dimensions, most notably HP's OpenView Dynamic Netvalue Analyzer [12]. On the other hand, such systems have to be purchased and maintained, and due to their involved high capital and operational expenditures, over-provisioning of network capacity till the point that having exact knowledge of the traffic matrix becomes unnecessary is considered as an easy and cost effective solution by network operators.

A. Our Contribution

We believe that bypassing a single IP link with an optical bypass configured on a separate physical port has the advantage over every other similar method, because the latter usually have large impact on the overall IP network operation. Another advantage is that IP routing is not impacted, since the established link is not advertised. Also configuration effort is limited to the two end points of the bypass instead of all routers along the path of the ingress-egress tunnel, thus reducing the configuration errors. In addition, the measurements we use in our approach are simple SNMP link counts, which do not require any monitoring equipment. The routers are only required to support policy based routing, which is a standard feature. The optical technology in place is required, which is a common feature, since the core IP link ports are based on optical transmission technologies. Due to the relatively low port sizes necessary for the bypasses (i.e., transmission speeds order of magnitude lower than that of the network traffic measured) and due to the fact that there is no need for any measurement infrastructure, we assume that our approach provides a cost efficient solution overall.

Establishing optical bypasses to measure OD flows is new, to the best of our knowledge. It should be noted, however, that the problem of bypass location appears quite similar to the problem of flow monitoring location, which has been studied before. However, direct flow monitoring decreases the router's performance, so that ISPs usually only sample the traffic, which has impact on accuracy [13]. In contrast to sampling, our method *directly measures* the OD flows using the SNMP link count on the bypass, so that an extrapolation from samples is not necessary. We also note here that the optimization of the locations of optical bypasses is different from the location optimization for the purpose of OD flow measurements, due to the
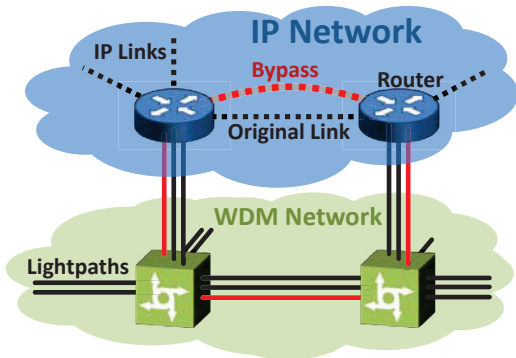
Fig. 1: Optical bypass in IP-over-WDM networks today.

fundamentally different objectives [3].

This paper also proposes the concept of traffic bounds to improve the utilization of the bypasses, which is novel. A method to generate optimum upper and lower bounds on OD flows was presented in [14], where the authors defined the problem as an ILP model, which has been shown effective for smaller size networks. For larger networks, however, as in our sample Cost266 network with 1332 OD flows, this method would result in 2664 separate optimization problems, which would be impractical. Compared to linear programming, our approach to derive traffic bounds and thus improve the accuracy is computationally inexpensive, since the proposed algorithm terminates after a few iterations (i.e., we have not observed computation times longer than three seconds on an off-the-shelf desktop computer even for the largest tested topology).

III. REFERENCE ARCHITECTURE

A. Creating Optical Bypass

In the network architecture studied, we assume that IP routers are co-located with optical switches, and an optical circuit can be established dynamically between the two underlying optical switches and used to create a link between the corresponding routers (Figure 1). This dynamic circuit can be setup automatically by the mechanisms of the control plane. Every time a new connection is created, it is advertised in the routing protocol as new IP link, which can cause routing instabilities. We avoid the routing advertisement, in our approach we assign an OSPF link cost to the bypass higher than the original IP link, which assures that the original link is always preferred over the bypass for IP routing. For the measurement on the bypass, explicit forwarding rules are created at the ingress router in order to offload specific traffic flows onto the optical bypass. It should be noted that in case where a highly utilized link is bypassed for measurements, the bypass itself may affect the traffic volume of the re-routed OD flow. We therefore limit the study to the cases of moderately loaded links.

For every OD flow that we aim to measure on a bypass, we need to define an access control list (ACL) such that the OD flow in question can be distinguished

from the remaining traffic. The ACL has to specify the address range of the source IP according to the ingress router and the address range of the destination IP according to the egress router of that OD flow. The management of the ACL can be done in a centralized fashion, where a server can be used to collect the necessary information from the ingress/egress routers. Note that the ACLs may become complex due to the fact that ingress/egress routers generally handle a lot of network prefixes. We afterwards configure a routing policy at the bypass ingress router that appoints the bypass port for all packets matching the ACL. In this way we provide that only packets of that specific OD flow are transmitted via the bypass, such that a SNMP link count function can be used for the measurement. As a result, a network operator does not have to use more than one routing policy at the same time in the same router for our approach. In this way, the overhead in packet processing is limited to one additional header field lookup (the source address), which justifies the usage of ACLs.
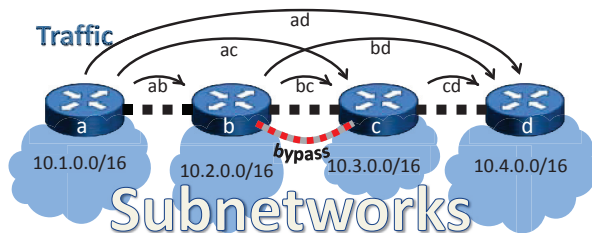

Fig. 2: Bypass example: the bypass parallel to link bc allows to measure flows ac, ad, bc, and bd.

B. The Measurements

Figure 2 shows an illustrative network using bypass-based measurements. Here, the topology includes four routers a, b, c and d, and for simplicity we only consider the traffic into the rightward direction with the total of six OD flows. The corresponding ill-posed linear system for the traffic matrix of this network is shown in Figure 4. It is easy to see that the most beneficial bypass to setup is from b to c, since the original IP link carries the most OD flows (f_{ac}, f_{ad}, f_{bc}, and f_{bd}). As shown in Figure 3, for each OD flow on the original link, we set up one access control list (ACL) on the bypass ingress router (here: router b).

Figure 3 shows how the routing table of the bypass ingress (Router b in Figure 2) changes over time: The first table shows the IP routing without bypass. After the bypass is setup, we sequentially use the ACLs 1, 2 and 3 to reroute all OD flows separately over the bypass. Afterwards, the bypass can be decommissioned and the routing is reset to its original state. The bypass capacity can be dimensioned using the bounds tightening method explained in Section IV-A.

Figure 4 illustrates how the traffic values measured can be used to increase the rank of the linear system. Each measured OD flow adds a new row to the linear system, which in our small-size network example is
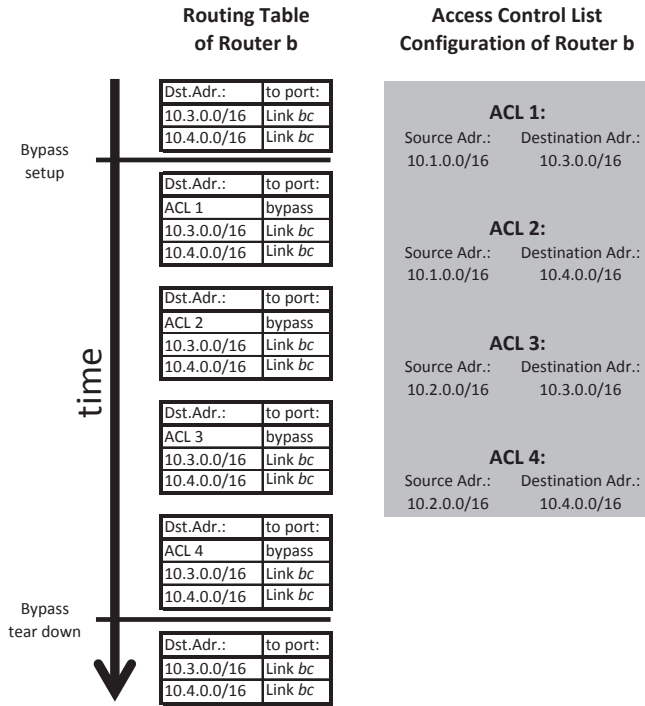
Routing Table of Router b

Access Control List Configuration of Router b

Bypass setup

time

Bypass tear down

Dst.Adr.:	to port:
10.3.0.0/16	Link *bc*
10.4.0.0/16	Link *bc*

Dst.Adr.:	to port:
ACL 1	bypass
10.3.0.0/16	Link *bc*
10.4.0.0/16	Link *bc*

Dst.Adr.:	to port:
ACL 2	bypass
10.3.0.0/16	Link *bc*
10.4.0.0/16	Link *bc*

Dst.Adr.:	to port:
ACL 3	bypass
10.3.0.0/16	Link *bc*
10.4.0.0/16	Link *bc*

Dst.Adr.:	to port:
ACL 4	bypass
10.3.0.0/16	Link *bc*
10.4.0.0/16	Link *bc*

Dst.Adr.:	to port:
10.3.0.0/16	Link *bc*
10.4.0.0/16	Link *bc*

ACL 1:
Source Adr.: 10.1.0.0/16   Destination Adr.: 10.3.0.0/16

ACL 2:
Source Adr.: 10.1.0.0/16   Destination Adr.: 10.4.0.0/16

ACL 3:
Source Adr.: 10.2.0.0/16   Destination Adr.: 10.3.0.0/16

ACL 4:
Source Adr.: 10.2.0.0/16   Destination Adr.: 10.4.0.0/16

Fig. 3: Routing table and access control list configuration of router b.

Original ill-posed Linear System:

on link ab:	$1\cdot f_{ab}$ +	$1\cdot f_{ac}$ +	$1\cdot f_{ad}$ +	$0\cdot f_{bc}$ +	$0\cdot f_{bd}$ +	$0\cdot f_{cd}$ =	36
on link bc:	$0\cdot f_{ab}$ +	$1\cdot f_{ac}$ +	$1\cdot f_{ad}$ +	$1\cdot f_{bc}$ +	$1\cdot f_{bd}$ +	$0\cdot f_{cd}$ =	54
on link cd:	$0\cdot f_{ab}$ +	$0\cdot f_{ac}$ +	$1\cdot f_{ad}$ +	$0\cdot f_{bc}$ +	$1\cdot f_{bd}$ +	$1\cdot f_{cd}$ =	44

Additional rows from the bypass:

possible measurements when bypassing link bc:

f_{ac} =	12
f_{ad} =	13
f_{bc} =	14
f_{bd} =	15

Resulting linear system …

on link ab:	f_{ab} + 12 + 13 =	36
on link cd:	13 + 15 + f_{cd} =	44

… can be solved completely:

on link ab:	f_{ab} =	11
on link cd:	f_{cd} =	16

Fig. 4: The increased rank due to the measurements on bypass bc allows to solve the linear system.

sufficient to solve the system completely. In larger systems, as we will see, the solution may not be complete unless the number of bypasses is increased.

All SNMP measurements on the same bypass have to be performed time sequentially, and the necessary uptime of the bypass depends on the minimum possible link count interval of the routers. Thus, within a longer measurement cycle, load values can vary considerably. Measuring for instance 100 flows sequentially on the same bypass with a minimum link count interval of 15 sec (where only every second interval can actually be used since every other interval is used to reconfigure the routing), the whole measurement process can take almost one hour. Note that as the link capacity required by the bypass is only a small fraction of the capacity of the original link (i.e., there's always only a single OD flow on it at the same time), the network operator can either leave the bypass permanently setup, or even provision multiple parallel bypasses to speed up the measurement (e.g., in case of high link utilization). Therefore, in our study, we assume that the traffic variations during a measurement cycle can be neglected. This assumption is reasonable since our OD flows represent a backbone's ingress-egress flows that typically carry highly aggregated (low-fluctuation) traffic. We also assume that the routers' minimum link count interval is relatively short.

IV. BYPASS CAPACITY AND PLACEMENT

So far, we have assumed that the bypass capacity is sufficient to offload and measure the largest OD flow on the corresponding IP link. However, since the de-

mands of the individual OD flows are unknown before measurement, the capacity of the bypass is likely to be over-dimensioned. In this section, we present an effective way to derive bounds on OD flows as well as an optimization method for bypass placements.

A. The Bypass Capacity by Tightening Traffic Bounds

We now propose a simple, yet highly effective method to derive upper and lower bounds on OD flows in the network, which in turn can help us to determine the required bypass capacity. While upper bounds on OD flows are necessary to avoid excessive over-provisioning of the bypasses , the method described here also calculates lower bounds. In this way, the method can independently provide solutions for a few OD flows, which can sometimes help us to find an OD flow where the gap between upper and lower bound can be closed completely. Figure 5 illustrates the idea.
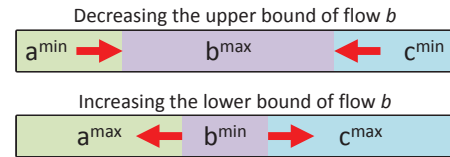
Decreasing the upper bound of flow *b*

$a^{\min}$ → $b^{\max}$ ← $c^{\min}$

Increasing the lower bound of flow *b*

$a^{\max}$ ← $b^{\min}$ → $c^{\max}$

Fig. 5: Tightening the bounds of OD flow b.

Based on Figure 5, let us assume a certain measurement value of the link load m, and from the routing information, we know that OD flows a, b and c contribute to that load, i.e., $m = a + b + c$. This is depicted in Figure 5, where m represents the width of the bar. In our notation, $x^{\max}$ denotes the upper bound, while $x^{\min}$ denotes the lower bound of x. We can now use the knowledge on the lower bounds of a and c to improve the upper bound of b: In case $b^{\max} > m - a^{\min} - c^{\min}$, we set $b^{\max} = m - a^{\min} - c^{\min}$. As shown in the figure, the method also works the other way around, i.e., by improving the lower bound of b with the upper bounds of a and c. Initially, all

lower bounds are set to zero, while all upper bounds are set to the lowest measured traffic value along the routing path of that flow. (Obviously, an OD flow can not be larger than any measurement it contributes to.) After initialization, the whole algorithm iterates over all measurements, while per measurement, it iterates over all OD flows contributing to that measurement. The algorithm stops when no further improvements can be achieved, which is – from our observations – after less then ten iterations.

Parameter	Meaning
Links	Set of all links the network
Flows	Set of all OD flows the network
n	Number of bypasses
u_λ	Boolean parameter, true if OD flow λ is unknown
R_ν^λ	Boolean routing parameter, true if OD flow λ is routed via link ν

Variable	Meaning
b_ν	Boolean to determine if link ν is bypassed
x_λ	Boolean to determine if OD flow λ can be measured

TABLE I: Summary of Notation

B. Bypass Placement

In order to select the most beneficial links to bypass for OD flow measurement, we developed the following heuristic approach: Out of all the links of the network, n are chosen to be bypassed, so that the total number of measurable *unknown* OD flows is maximized. (Please note that *known* OD flows may be derived earlier from bounds tightening.) For choosing the n links, we developed a simple ILP model. The used parameters and variables are described in Table I. The objective function of the optimization problem is defined as:

$$\text{Maximize} \quad \sum_\lambda x_\lambda \cdot u_\lambda \quad (1)$$

which maximizes the number of measurable (and unknown) OD flows. In order to constrain the measurability x_λ of an OD flow λ, we demand

$$\forall \lambda \in \text{Flows}, \forall \nu \in \text{Links}: \quad x_\lambda \leq R_\nu^\lambda \cdot b_\nu \quad (2)$$

to assure that a measured OD flow λ is routed via a bypassed link. The maximum number of bypasses is constraint by:

$$\sum_\nu b_\nu = n \quad (3)$$

Please note that while this ILP defines the optimum choice of bypasses for the maximum number of OD flows measured on that bypasses, it does actually not define the optimum choice of bypasses for the maximum number of *finally obtainable* OD flows. The reason is that a measured OD flow may or may not lead to the solution of another OD flow from the linear system $L = R \cdot F$, which is ignored in our ILP. That

is why we consider our bypass location approach a heuristic, rather than an optimization approach.

The measured OD flows are then added to the linear system $L = R \cdot F$ as new rows in order to increase its rank. The number of finally obtainable OD flows is usually still considerably higher than the number of measurements. In order to calculate those additional flows, we row-reduce the linear system with the Gaussian Elimination algorithm [15]. This is numerically challenging due to the enormous size of the system, leading to either impractical large numbers or rounding errors. However, since the system is sparse (i.e., most values are zero), we can tackle the problem with the Markowitz Pivot Strategy to keep the Gauss-Jordan algorithm numerically stable and to preserve sparsity of the system [15].

V. Performance Evaluation

The base network topology used is the Cost266 European network (37 nodes, 57 link, 1332 OD flows). In addition, to determine the impact of the network size on the performance outcome, we tested our approach with two additional network topologies: the Atlanta network as an example for a smaller network (15 nodes, 22 links), and the TA2 network as large scale topology (65 nodes, 108 links). The traffic matrix is randomly generated, and is only used to compute all link loads; however, all steps of our approach, we assume no knowledge on the traffic matrix. Since all steps of our approach consider only on the *number* of obtained OD flows, the absolute traffic values have effectively no influence on the results obtained. We use destination based shortest path routing for all OD flows. This means 1) that only shortest paths are allowed, and 2) that two flows with the same destination, once they are merged at a common router, can not split anymore[1]. Regarding the order of operations in our method, it is important that the linear system is row-reduced before bypass placement, because there are OD flows that can already be derived from it, and for bypass placement those flows don't need to be anymore considered. After the optimization, the row-reduce operation is applied again, leading to additionally solved flows. Due to the short computation time, we perform bounds tightening before each step and at the very end. All operations of our approach, including the bypass placement optimization and the matrix row-reduce operations, were computed on an off-the-shelf desktop computer in less than one minute. For bypass placement optimization we used the Gurobi solver [16], and the used topologies are taken from the SNDlib library [17].

With only 7 bypasses in the Cost266 network, it is possible to obtain the exact demand of 71% of all OD flows, either by direct measurement or by deducing them from the linear system. We assume that using this data as side information for traffic matrix estimation will lead to estimates with low errors. How the

---

[1]This is exactly what happens in a common IP router, since packet forwarding is based on the destination address.

performance of our approach scales with the number of bypasses is shown in Figure 6, where it can be seen that adding bypasses lets the performance increase fast, as long as there are still enough links carrying a high number of flows. With more bypasses already in the system, the further improvement by adding more bypasses is declining. Finally, by bypassing 33 out of the 57 links, we can obtain all OD flows in the network. The number of OD flows that can be measured on the bypasses is not independent of routing scheme used in the network. In this regard, when more flows are routed over only a few links, it is easier to capture a high fraction of the total traffic with only a few bypasses. However, due to our strong routing constraints (i.e., shortest path, destination based forwarding), the choices left to make during the path computation have no significant impact on the results.
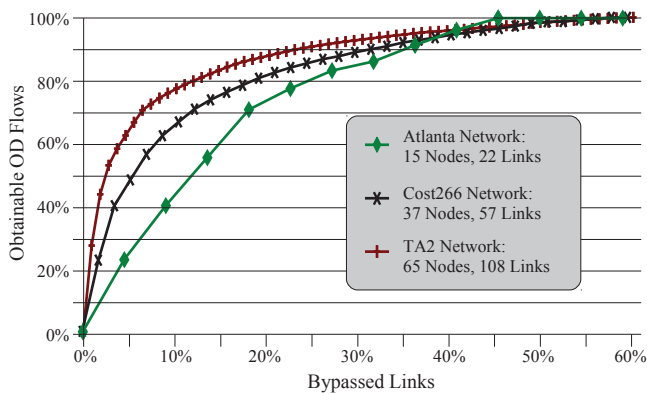

Fig. 6: Performance comparison for diff. network sizes.

Figure 6 shows how the performance of our approach scales with the number of bypasses in various topologies. Our findings in this evaluation are that the bypass method works well in all three compared network topologies. The results differ in regard to the amount of necessary bypasses towards the full solution (i.e., all OD flows obtainable); here, we can see that in the largest network we need to bypass 60.2% of the links, in the mid-size network it's 57.9%, while in the small network it's only 45.5%. On the other hand, we can also see from the diagram that having a small amount of bypasses provides better results in larger networks.

Number of used Bypasses:	5	10	15
Q with untightened bounds:	30.2	30.7	32.4
Q with tightened bounds:	11.8	5.5	3.3

TABLE II: Impact of Bounds Tightening.

We expect that prior knowledge on upper and lower bounds of OD flows (e.g., derived from flow sampling) can considerably improve the performance of our algorithm, but in this study we assume that all OD flows are completely unknown at the beginning. Although bounds tightening shows no impact on the

number of obtainable OD flows, we found it useful to dimension the bypass capacities instead, while another advantage of tightened bounds can be more important: For a successional traffic matrix estimation (e.g., using the tomogravity model), our bounds are of high value for more accurate results. Table II shows the average difference between upper and lower bounds of unsolved OD flows relative to the actual traffic demand in the Cost266 network, measured as ratio $Q = \sum_f \Delta_f / \sum_f \lambda_f$ with Δ_f as difference between upper and lower bounds of flow f and λ_f as actual traffic value of flow f. Although the impact of our bounds on various traffic matrix estimation is out of the scope of this paper, these results suggest that we can considerably decrease estimation errors.

VI. CONCLUSIONS

With this paper, we believe to have advanced the knowledge of accurate computation of IP traffic matrices by proposing a novel approach to measure individual OD flows using optical bypass. Instead of using expensive monitoring infrastructure, we propose to use policy based routing and dynamic circuit setup, two features that are likely to be available in the core networks. The proposed method does not impact the IP routing in place, and our results show that a few bypasses are sufficient to obtain actual traffic demand values for most of the entries in the traffic matrix. An important result of our analysis is that our approach performs well for all tested network sizes, but with increasing network size the relative amount of necessary bypasses to obtain the full traffic matrix increases. Finally, we discussed how the tight upper and lower bounds on OD flows we computed can be used as constraints for a successional traffic matrix estimation in order to lower estimation errors. We leave the actual application of the computed bounds to traffic matrix estimation for future work.

REFERENCES

[1] K. Papagiannaki, N. Taft, A. Lakhina, *"A distributed approach to measure IP traffic matrices,"* SIGCOMM conference on Internet measurement 2004
[2] R. Inkret, A. Kuchar, B. Mikac, *"Advanced Infrastructures for photonic Networks,"* Extended Final Report of COST Action 266
[3] M. Chamania, M. Caria, A. Jukan, *"Achieving IP routing stability with optical bypass,"* Optical Switching and Networking, vol.7, issue 4, pp.173-184, Dec. 2010
[4] Y. Vardi, *"Network tomography: Estimating source-destination traffic intensities from link data,"* Journal of the American Statistical Association, v.91, n.433, pp.365-377, 1996.
[5] D. Lam, D.C. Cox, J. Widom, *"Teletraffic modeling for personal communications services,"* IEEE Communications Magazine, v.35, n.2, pp.79-87, Feb 1997
[6] Y. Zhang, M. Roughan, N. Duffield, A. Greenberg, *"Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads,"* SIGMETRICS 2003
[7] A. Nucci, R. Cruz, N. Taft, C. Diot, *"Design of IGP link weight changes for estimation of traffic matrices,"* INFOCOM 2004

[8] S. Raza, G. Huang, C.-N. Chuah, S. Seetharaman, J.P. Singh, *"MeasuRouting: A Framework for Routing Assisted Traffic Monitoring,"* INFOCOM 2010

[9] J. Sommers, B. Eriksson, P. Barford, *"On the Prevalence and Characteristics of MPLS Deployments in the Open Internet,"* SIGCOMM conference on Internet measurement 2011

[10] N. McKeown, et al., *"OpenFlow: Enabling Innovation in College Networks,"* Whitepaper, www.openflow.org

[11] Cisco, *"NetFlow Performance Analysis,"* Technical White Paper, May, 2007

[12] J. Altmann, L. Rhodes, *"Dynamic Netvalue Analyzer - A Pricing Plan Modeling Tool for ISPs Using Actual Network Usage Data,"* IEEE WECWIS 2002

[13] G.R. Cantieni, et al., *"Reformulating the Monitor Placement Problem: Optimal Network-Wide Sampling,"* Conference on Information Sciences and Systems, 2006

[14] A. Gunnar, M. Johansson, T. Telkamp, *"Traffic Matrix Estimation on a Large IP Backbone - A Comparison on Real Data,"* SIGCOMM conference on Internet measurement 2004

[15] Timothy Vismor, *"Matrix Algorithms,"* March 2012

[16] Gurobi Optimizer 3.0, http://www.gurobi.com/

[17] SNDlib library, http://sndlib.zib.de