Traffic Engineering using Segment Routing and Considering Requirements of a Carrier IP Network

Timmy Schüller°•, Nils Aschenbruck°, Markus Chimani°, Martin Horneffer*, Stefan Schnitter•

°University of Osnabrück, Institute of Computer Science
Wachsbleiche 27, 49076 Osnabrück, Germany
Email: {tschueller, aschenbruck, markus.chimani}@uos.de

•Detecon International GmbH
Sternengasse 14-16, 50676 Köln, Germany
Email: {timmy.schueller, stefan.schnitter}@detecon.com

*Deutsche Telekom, Technical Engineering Center
Dahlweg 100, 48153 Münster, Germany
Email: {martin.horneffer}@telekom.de

***Abstract*—Internet Service Providers (ISPs) are challenged by increasing traffic demands. Advanced Traffic Engineering (TE) is one way to overcome this challenge. Segment Routing (SR) is a relatively new approach for TE. To decide whether SR is a good approach for deployment in carrier IP backbone networks, it has to show its benefits in real-world scenarios and still needs to be feasible from the network operation and management point of view. In this paper, we analyze traffic data from a European tier one backbone network from 2011 to 2015. The total traffic increases significantly throughout that period. We analyze geographic differences to select representative traffic peak times as reference scenarios for an evaluation of TE using SR for real-world topologies and traffic demands. Finally, we extend existing SR formulations to consider requirements from network operation and management. Our evaluation results show that SR yields close to optimal results while still being deployable with reasonable effort.**

I. Introduction

Nowadays, customers do not only download content to their home computers. They stay connected while being mobile. They want to access and synchronize their data and stream music or videos from their phones, laptops, or other mobile devices. As a result it is not surprising that the amount of Internet traffic is increasing. The Cisco Forecast [4] predicts that the annual global IP traffic will surpass the zettabyte threshold in 2016 and even double it in 2019. This trend probably will continue as technology advances. Compensating the increasing demands of more and more traffic is a major challenge for Internet Service Providers (ISPs). There are two ways to tackle the problem and usually both of them are necessary. First, the network can be physically expanded, which is expensive. Second, the utilization of the available resources can be optimized with the use of Traffic Engineering (TE).

In this paper, Segment Routing (SR), a strategy that recently gained popularity in TE, is implemented and tested using real world data from the backbone of a European tier one ISP. The goal is to examine whether SR is a feasible approach for deployment in carrier IP backbone networks. To be able to perform effective TE, it is important to focus on the most challenging situations. An intuitive choice for a peak hour, for example, would be somewhere in the evening, when people come home from work. However, for a backbone network that has nodes located in three continents, different time zones may lead to different peak hours.

After a set of topologies and representative traffic data is selected, SR for TE has to be evaluated against the optimum (Multicommodity Flow Problem, MCF [15, Chapter 4.4]) as well as the state of the art approaches. We base this evaluation on an SR approach first introduced in [3]. This evaluation will show us for our backbone network how close we can come to the optimum using SR as well as how large the difference to the current state of the art is.

However, beside pure TE there are also requirements from the network operation and management point of view. To consider this, we extend the original Segment Routing formulation. In practice, without using centralized control mechanisms like Software-Defined Networking (SDN) [11], it can be cumbersome to configure many SR paths. Thus, our Tunnel Minimization Extension (TME) attempts to minimize the number of segments used. For this purpose, we call a path that uses at least one intermediate segment a SR tunnel. To match even stricter practical assumptions, our Tunnel Limit Extension (TLE) prohibits splitting up traffic demands. For both extensions, we show the evaluation results comparing them with each other as well as with earlier results.

The paper is organized as follows. At the beginning, a brief overview on background knowledge including the related work is presented (Section II). Based on this we describe the open challenges in detail (Section III). Then, the traffic data from an IP backbone network is analyzed (Section IV). Using the derived reference scenarios, we evaluate TE based on SR. Next, extensions that cover real-world constraints are introduced and evaluated (Section VI). Finally, we conclude the paper and point out topics for future work (Section VII).

II. Background

This section provides a brief overview on background knowledge and related work.

ISBN 978-3-901882-94-4 © 2017 IFIP

A. Internet Traffic Analysis

The topic of Internet traffic analysis has been of interest since the origin of the Internet. In [16], for example, delay experiments between remote hosts on the ARPANET, the MILNET, and others were performed to improve the Transmission Control Protocol (TCP). Often, e.g. in [5], the focus is on the composition of Internet traffic, breaking down the contribution of different protocols and applications, or the distribution of different packet sizes and data flow lengths. Regarding the actual growth of traffic, Coffman and Odlyzko [6] did an interesting prediction: They debate the question whether there is a Moore's Law for Internet traffic. The conclusion is that "Internet traffic is likely to continue doubling each year for the next decade or so" ([6], p.89).

A more recent report published on behalf of Alcatel-Lucent includes various data sets and predictions, such as the Cisco Forecast or the Minnesota Internet Traffic Study (MINTS). It shows the increase of Internet traffic from about 1990 to 2015 and approximately agrees with the results from Coffman and Odlyzko [7]. It does, however, not give a view on how the traffic is shaped over a typical day, but rather focuses on capacity and limits of optical communication network traffic.

To the best of our knowledge, there is no other recent paper that analyzes the traffic growth in a tier one ISP backbone network. The question which time of the day is representative and is appropriate for TE also seems to be open.

B. General Traffic Engineering

There are many ideas and strategies on how to optimize routing in a network, some of which are presented below. Generally, TE is performed on a specific network topology. Traffic demands enter the network on ingress nodes and exit the network through egress nodes. Theoretically, any router in the network can be an ingress node, an egress node, or even both.

1) Metric Optimization: Most Interior Gateway Protocols (IGPs) like Open Shortest Path First (OSPF) [17] or Intermediate System to Intermediate System (IS-IS) [18] rely on simple shortest path routing. Manipulating these shortest paths by adjusting the IGP metrics, as described in [10], is an elegant solution that can be deployed easily. Even if done in a clever way, this approach still follows shortest paths. Its effectiveness, therefore, is limited, as discussed in more detail later on. Even though it is not the optimal solution, metric optimization is the state of the art for many ISP's due to its simplicity.

2) Multicommodity Flow (MCF) Problem: The Multicommodity Flow problem is a network flow problem in combinatorical optimization. Given a graph with edge capacities and multiple independent commodities that are to be transported through the graph. The task is to find paths for the commodities, such that the flow is maximized. This problem is well understood and can be solved efficiently with linear programs when fractional solutions are allowed [8]. The basic problem itself may only consider the maximum flow for multiple demands, but can also be formulated to minimize the maximum link utilization as shown in Problem 1 later on.

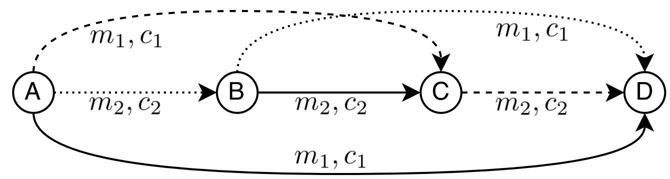

Figure 1: Constructed worst-case topology for 2-SR.

3) Other Strategies: Several other strategies can be applied for TE. With Multiprotocol Label Switching (MPLS) [20] and Resource Reservation Protocol (RSVP) [2] explicit TE tunnels can be deployed. Tunnels can also be used to provide backup routes in failure cases. In contrast to a metric optimization, this technique requires the use of additional network protocols to manage these tunnels. Theoretically, it can be used to deploy MCF results. It is, however, a very time-consuming task to deploy and maintain arbitrary tunnels for each pair of ingress-egress nodes. This is why only few operators have adapted this methodology.

All previously presented strategies tend to be beneficial for a medium time frame only. A metric optimization is computed, for example, once a day at midnight. But technologies like SDN [11], where the network is controlled in a centralized manner, enable the possibility for a short-term or even online TE. In [1], TE is shown to be effective even when a SDN is deployed incrementally.

C. Traffic Engineering using Segment Routing

This paper focuses on Segment Routing as a TE strategy, first introduced in [9]. With SR, a demand can be directed through certain segments to achieve more efficient routing. To reach a specific segment, the IGP is consulted. Within the scope of this work, a segment always represents a node in the network. In general, segments could also be links or services, such as Virtual Private Network (VPN) services. Defining a limited number of intermediate nodes for demands is much simpler to deploy than defining arbitrary paths as done in MCF. Recently, Bhatia et al. [3] published linear programming approaches that implement different variants of SR. Their formulations all rely on the claim that picking only one intermediate node already leads to near optimal results. They call this limited version *2-Segment Routing (2-SR)*. However, their evaluations are done on randomly generated topologies and do not necessarily reflect the situation in real networks. The first of the programs in [3] acts as a basis for the extensions implemented here. Our extensions cover a more complex problem than just to minimize utilization of the network, which is the main focus of current literature on SR. In addition to load optimization, our extensions satisfy additional real-world constraints.

Other approaches to Segment Routing can also be found in the literature. For example, in [12] a hybrid constraint programming framework was developed. Unlike in [3], they also evaluate their framework on real network topologies. The method, however, still focuses on solely optimizing maximum link utilization.

III. Problem Statement

This section describes the problems addressed in this paper in detail. First, we motivate the necessity of performing the evaluations on real topologies and traffic by constructing a worst-case topology. Second, we describe real-world constraints coming from network operation and management.

A. The Importance of Real-World Data

While the approach in [3] shows good results for randomly generated topologies and traffic matrices, it is unclear if this holds true for real-world data. One can easily construct a worst-case example where 2-SR fails to perform. Figure 1 shows a worst-case topology consisting of only four nodes and six edges. There are two types of edges. Type 1 edges are curved. They are defined by a low metric m_1 and a low capacity c_1. Type 2 edges are straight. They have a high metric $m_2 \geq 3m_1$ and a capacity of $c_2 > c_1$. Using these edges, a demand D is to be routed from A to D. To achieve this, the topology offers four possible paths:

1) One direct path from A to D using a type 1 edge. This is the shortest path.
2) One path that uses only type 2 edges.
3) Two paths that use both edge types. They are visualized with dotted and dashed edges respectively.

Multicommodity Flow (MCF) will balance the demand between all four paths. This behaviour yields a maximum link utilization of $X/(C+2)$, as X units exit node A through two type 1 edges and one type 2 edge. The naive shortest path approach only uses the first path, terminating with a maximum link utilization of X. To make it hard for 2-SR, the topology is constructed in such a way that the second path can never be selected by 2-SR. Because every SR path consists of two concatenated shortest path, somewhere along the path a type 1 edge will get chosen. At the end, there are three possible tunnels, which are weighted equally. This results in a maximum link utilization of $X/3$, which is still better than naive shortest path routing. But, other than the utilization in the case of MCF, it is independent of C. As a result, 2-SR can be arbitrarily bad in comparison to MCF.

However, this is a very theoretic example, as it contradicts the idea of metric design. In practice, metrics are usually set anti-proportional to capacity. Also, it is neither appropriate to pick a randomly generated matrix or topology, nor to randomly pick real matrices. It is worthwhile to deliberately choose representative matrices. The main challenge is to pick a subset of representative samples to run optimizations on. Given that our tier one network has nodes located in three continents, this is potentially a difficult task. Different time zones lead to different peak hours.

B. Additional Real-World Constraints

Beside engineering the real-world traffic in an optimal way, it is also important to look at the constraints from the network management and operation point of view. For our network, there are two important constraints: (1) Every SR tunnel has to be deployed and maintained. Therefore, depending on the operational model, it is desirable to keep the number of SR tunnels to a minimum. (2) All SR formulations in [3] allow splitting up demands with arbitrary factors. This can lead to problems in practice, as typical routers (for example routers running JUNOS [14]) only support splitting each demand into up to 16, 32, or 64 equally-sized parts. Of course (2) also directly impacts (1). If a demand is split in k fraction (say $1/32$), it will result in k tunnels to be deployed.

IV. Carrier IP Network Traffic Analysis

In this section, we describe and analyze the traffic data set from a European tier one carrier IP backbone network. This analysis focuses on examining the traffic growth and on finding a representative peak hour for TE.

A. Measurement Architecture

The analysis as well as all evaluations in the remaining part of this paper are based on measured traffic matrices that contain information about how many kbit/s were delivered between any two routers within the traffic-engineered sub-topology of the network. Every day is recorded in 96 matrices; each matrix captures the average traffic during a 15 minute window. The measurements were done partially by reading MPLS Forwarding Equivalence Class (FEC) counters and partially with estimation techniques described in [21]. This combination ensures that the measurement error stays within 10%. The original topology is simplified to combine edge routers at each Point of Presence (POP) to one virtual node. This process results in a virtual topology with 100–150 nodes and a network density of approximately 5% compared to a complete graph. For this analysis we obtained access to matrices of the virtual topology of one workday and one weekend day per week from May 2011 to August 2015.

B. Growth of Internet Traffic

To approximate the growth of Internet traffic over time we divided the dataset by year. It should be noted that the dataset starts in May 2011 and ends in August 2015, hence both years are incomplete. Nonetheless, they give an insight into the developments during that time.

Figure 2 visualizes the results. For each year and for Mondays and Sundays respectively, it shows the average traffic of all quarter hours. It is very clear that traffic being routed through the considered backbone network increases each year. The biggest differences are recorded around peak hours, while the differences at night are relatively small. The evening hours form a clear peak since 2014, especially on Mondays. In contrast, the curve was quite flat in 2011 and 2012. Apart from this detail, the shape of the curves stays similar over the years. The establishment of a very pronounced peak hour can be seen as a sign of declining popularity of peer to peer transmissions. Consumers appear to rather stream content live after coming home, which causes the peak, instead of downloading it over night or while away. Transmitting contents using peer to peer services was very popular about 15 years ago [22], while streaming services seem to be more important nowadays [23].

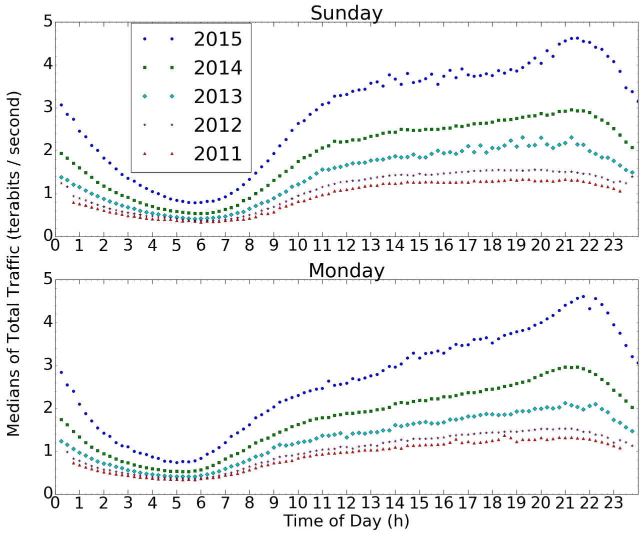

Figure 2: Comparison of total traffic in years 2011 to 2015 on Sundays and Mondays in tbit/s.

This behaviour can also be found in other networks, e.g., the Google backbone network (cf. [19], p.21).

C. Selecting a Peak Hour

When looking at Figure 2, choosing a peak hour can be done by looking at the global maximum for each year. Overall, the traffic curve shows an expected behaviour that approximately follows everyday's life. To be able to speak about exact times, all dates in this paper refer to the Central European Time (CET) and the Central European Summer Time (CEST), respectively.

The traffic low point is at about 5:00h–6:00h at night. From that point on the traffic slowly rises until the peak just before 22:00h. As some people have work to do that is not related to the Internet, the curve on weekdays is climbing slower than on the weekend. The peak in the after hours stands out a bit more in contrast to the more flat weekend evening. This can be seen very well for the years 2014 and 2015. Following these observations, the best time to pick a representative traffic matrix for a day is around 21:45h. This holds true for weekdays as well as weekends. While the total traffic of a whole day looks to be higher at weekends, the traffic peak is roughly on the same level.

D. Impact of Different Regions on the Choice of only one Peak Hour

While picking a peak hour simply at the point where the network delivers the highest amount of data seems reasonable, it could be problematic running optimizations only with this choice. If the network has nodes located in multiple time zones, which is the case in the network that is examined here, it could happen that the TE optimizations result in bad configurations for those other regions. Thus, the impact of traffic in other time zones has to be considered.

The first graph in Figure 3 captures every Monday contained in the dataset in 2014, visualized by boxplots. We only show traffic on Mondays, since the traffic does not differ significantly from that on Sundays. A single boxplot contains all traffic aggregations of the corresponding 15 minute window. The observations focus on the data from 2014, as it is more complete than 2015 and relatively recent. Three major regions were seperated to analyze flows between routers within the same region (intracontinental), as well as in- and outbound traffic of a region. In the first plot, only the intracontinental traffic is considered. Asia and America contribute much less to the overall traffic load than Europe.

The different time zones can roughly be observed in the plot. Compared to Europe, the drop in traffic at night occurs later in America at about 12:00h European time and earlier in Asia at about 23:00h European time. The intercontinental traffic can be seen in the second and third plot of Figure 3, where the outbound and inbound traffic is displayed respectively. In both directions, Asia has only a very small amount of traffic. The other two regions show characteristics that mirror vice versa. For inbound traffic, Europe has a traffic curve that is very similar to the intracontinental traffic, only on a lower scale; America has a more constant curve with a little more traffic than Asia.

Following these observations, most of intercontinental traffic seems to be delivered from the US to Europe. Given the fact that the time-dependant changes of American outbound traffic also mimic the European curve and time zone, it is likely that the nodes located in America are mostly servers and not customers. For the decision of picking a representative peak hour, this means that picking one for a day suffices. If needed, however, a second peak at for example 4:00h European time, where a peak hour would be expected for American traffic, can be considered.

E. Selecting a Reference Scenario Set

Following the previous observations, the dataset can be reduced to a manageable few samples, to be used in further evaluations. It is not possible to aggregate matrices of one year to use the average traffic loads for each link, because the topology of the network changes significantly during the measurement period. Thus, a few sample days per year are selected. It is not relevant to see how the network performs at night time or in the morning; instead it is appropriate to pick a matrix at the previously determined main traffic peak hour.

Four matrices were selected for each year. One in March, June, September, and December at 21:45h. Due to holes in the dataset, the data points of September 2012 and for March 2015 are missing. For the remaining part of the paper, we use these 16 traffic matrices as our reference scenario set.

V. Segment Routing Performance Evaluation

Given our set of reference scenarios, the goal is to test the network performance with a focus on the highest link utilization. Three TE approaches are considered: (1) A naive shortest path routing simulation including Equal-Cost Multi-Path (ECMP) was implemented to get an upper bound for network utilization. Since most ISPs deploy protocols like

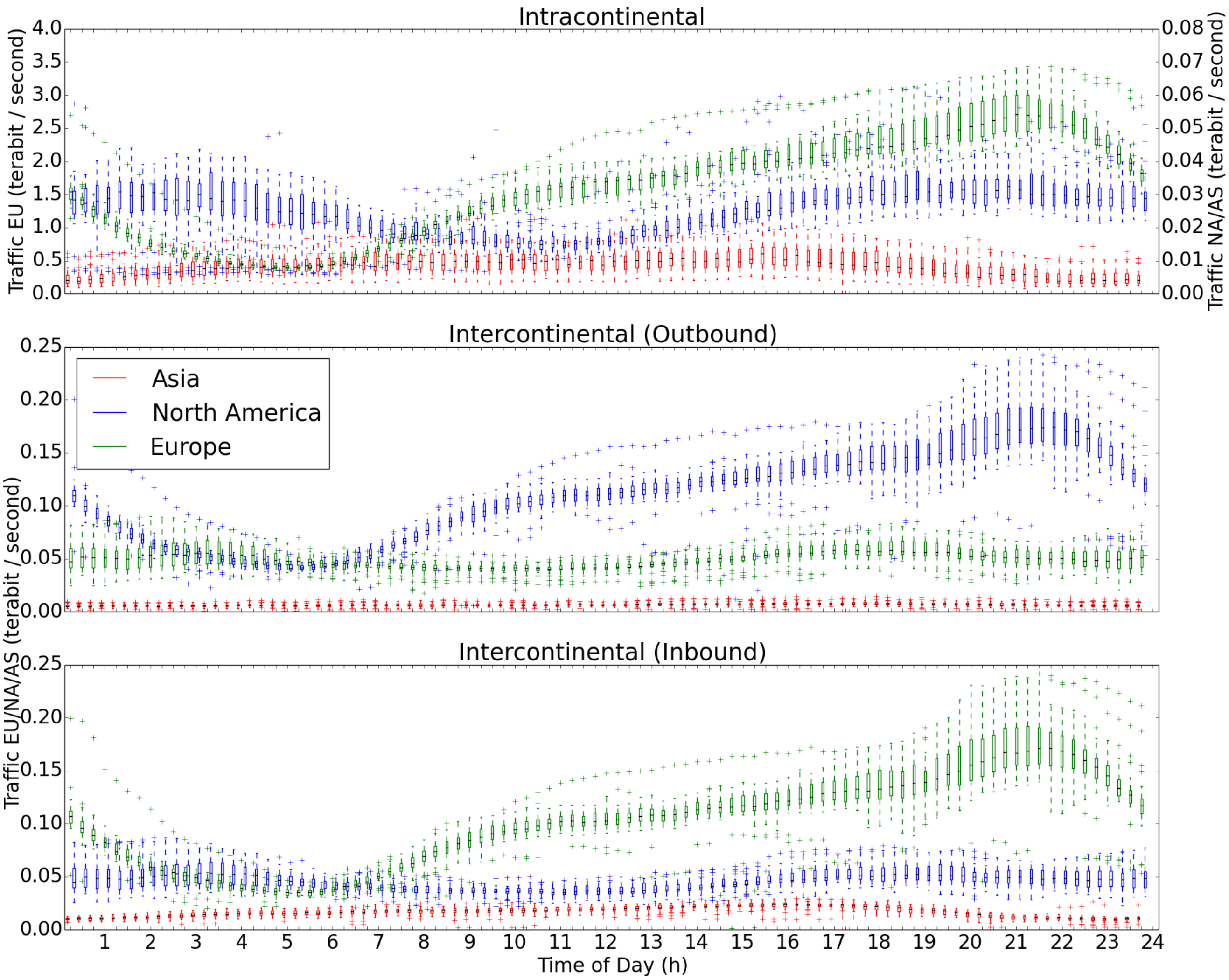

Figure 3: Intracontinental, Outbound, and Inbound Traffic in 2014 per region on Mondays in tbit/s.

OSPF or IS-IS as discussed earlier, this simple simulation comes pretty close to currently deployed technology. (2) In contrast to the naive shortest path routing, a formulation for the Multicommodity Flow is used to give a lower bound for network utilization. When limiting the possibilities of TE approaches to only redistribute traffic, MCF will result in an optimal situation, as arbitrary paths are allowed for each commodity. (3) The linear programming approach for SR as formulated in [3] is used to examine the performance of SR in real-world networks. If the results of SR presented in [3] for artificial topologies can be validated, the results are expected to be close to the ones of MCF.

A. General Implementation Details

All three algorithms require detailed knowledge about the traffic that is to be routed in advance. They can be classified as traffic matrix aware algorithms. The same set of traffic matrices, as described in the previous section, was used for the three algorithms.

Parsing of the data and further precomputations were implemented in C++. IBM/ILOG CPLEX [13] was used to solve the Linear Programs (LPs) for the second and third approach. Implementation details for all three algorithms and a comparison of the results are presented below.

B. Shortest Path Routing

Routing traffic along its shortest path is a widely used routine in network routing. It can be implemented easily and is rather efficient and practical in deployment. First, all shortest paths are computed for all pairs in the network. Second, all traffic demands are simply routed along those shortest paths. If there is more than one shortest path, the traffic is shared equally between all shortest paths. Capacity limits of links are ignored to keep the problem as simple as possible and can, therefore, theoretically exceed 100%. It should be noted that the metrics used to determine the shortest paths are already optimized. Thus, this simple approach represents more than an upper bound–it is the current state of the art.

$$
\begin{aligned}
&\min\ \theta \\
&\text{s.t.} \quad \sum_{j\in V} f^{s,d}(i,j) - \sum_{j\in V} f^{s,d}(j,i) \;=\; \begin{cases} t(s,d), & \text{if } s=i; \\ -t(s,d), & \text{if } d=i; \\ 0, & \text{otherwise.} \end{cases} && \forall (s,d)\in K, \forall i\in V && (1) \\
&\qquad \sum_{(s,d)\in K} f^{s,d}(i,j) \;\le\; \theta\, c(i,j) && \forall (i,j)\in E && (2) \\
&\qquad f^{s,d}(i,j) \;\ge\; 0 && \forall (s,d)\in K, \forall (i,j)\in E && (3)
\end{aligned}
$$

Problem 1: MCF formulation that minimizes maximum link utilization.

$$
\begin{aligned}
&\min\ \theta \\
&\text{s.t.} \quad \sum_k x_{ij}^k \;\ge\; t_{ij} && \forall (ij) && (4) \\
&\sum_{ij}\sum_k g_{ij}^k(e) x_{ij}^k \;\le\; \theta\, c(e) && \forall e && (5) \\
&x_{ij}^k \;\ge\; 0 && \forall (ij) && (6)
\end{aligned}
$$

Problem 2: Traffic matrix aware (the traffic demands are known in advance) 2-Segment Routing formulation by [3].

C. Multicommodity Flow

As motivated above, MCF serves as a lower bound estimate. To minimize the maximum utilization of the network, the basic MCF can be formulated as shown in Problem 1 (cf. [15, Chapter 4.4]). The formulation is based on a Graph G with a set of nodes V, a set of edges E, and a set of traffic demands K. The variables of the linear program are $f^{s,d}(i,j)$, describing the amount of traffic from source s to destination d via directed edge (i,j). The objective is θ, which stands for the highest link utilization and is defined by the relation between the vaiables and capacity constants $c(i,j)$. The actual traffic demands are represented by constants $t(s,d)$. Equation 1 represents the flow preservation and demand satisfaction constraint. Equation 2 binds the variables to the respective capacity which is scaled by θ. To reach an optimized network utilization using arbitrary paths for each demand, θ is minimized as the overall objective.

D. Traffic Aware 2-Segment Routing

The basic traffic aware 2-SR approach is originally defined in [3]. Their LP formulation is rather straightforward and can be found in Problem 2.

The overall objective, symbolized by θ, is to minimize the load on the connection with the highest utilization. This describes how well the network is able to handle the occuring traffic given in t_{ij}. The variables in this program, x_{ij}^k, define the absolute amount of traffic from node i to j that is routed through intermediate node k. Similar to load sharing in shortest path routing, each demand can be split up and be routed along multiple different intermediate nodes. The weights for these sub demands, however, can be chosen arbitrarily. It could, for example, be beneficial for the overall objective to route 20% of a demand through intermediate node k and 80% through k'. Equation 4 assures that the traffic demands given by the matrix are all satisfied by sub-demands. For our purposes we simplify this equation by changing the equal or greater to an equal sign. Equation 5 ensures that all sub-demands are satisfied while limiting each edges capacity $c(e)$ according to θ. For each link, $g_{ij}^k(e)$ gives information about how much a link e will be utilized, given a uniform flow from i to j through k. The function also uses shortest path information, as the paths from i to k and from k to j are essentially two linked shortest paths. The values of g can be precomputed for all demands, intermediate nodes, and links and are, therefore, constants in the LP.

E. Evaluation Results

The results of the three algorithms for the selected dates are shown in Figure 4. The values for network utilizations in this and all following figures are normalized to the highest utilization result of shortest path routing. First of all, no significant increase in worst case utilization can be seen over time. Most of the values reveal a utilization of around 80% for shortest path routing and around 50% for the other approaches. This shows that the traffic demand at peak hours still has room for more traffic when SR or MCF is used. As the traffic demand increased over time, which was discussed in Section IV-B, this means that the network has been expanded in such a way that the growth could be compensated. This is backed up by the blue diamonds in the same figure. They depict the sum of the capacities of all links in the network topology. The increase in capacity approximately mimics the increase of traffic demands.

Further, we see that 2-SR works well with real world instances. Topologies as constructed in Figure 1 do not seem to appear or impact the network, which is probably due to the increased complexity and size that a real network has. The lower bound values provided by MCF are matched in all cases but one. This shows that choosing only one intermediate segment is sufficient for TE. The complexity of choosing arbitrary paths with MCF should be avoided in practice, as the performance gain is neglectable or in most cases nonexistent.

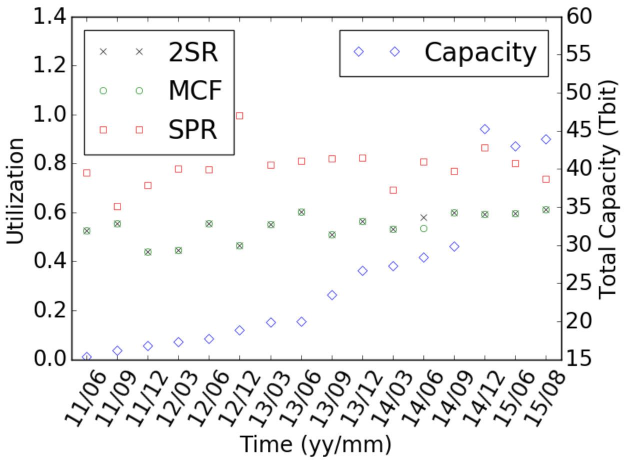

Figure 4: The left scale shows the utilization of the connection with the highest utilization for selected sample matrices as optimized by a LP. The right shows the cumulative capacity of the network.

VI. 2-Segment Routing Extensions

In the previous section, it was shown that 2-SR works well in a real network. However, the constraints described in section III-B are not considered yet. From a certain network management and operation perspective, the number of tunnels to be deployed should be as small as possible. When using the simple 2-SR approach, approximately 3000 tunnels are needed for each of our reference scenarios. In environments without a central controller that enables automation, each tunnel would have to be configured manually. Additionally, splitting up demands with arbitrary ratios is not feasible. Thus, there is a need to extend 2-SR considering both additional constraints.

A. Tunnel Minimization Extension (TME)

The minimization of the number of SR tunnels can be seen as a second objective. Multiobjective linear programming is difficult, but can be attempted by either weighting the objectives in a combined function, or by a progressive goal programming approach. For this use case, a combination of both techniques is used.

In a first optimization step, the maximum link utilization is minimized with the original 2-Segment Routing linear program described in Problem 2. The result is then included to a second optimization problem as equation 9. The complete formulation can be found in Problem 3. Equation 9 limits the change of the primary objective with a prespecified, fixed tolerance factor $\lambda \geq 1$. For example, $\lambda = 1.2$ allows θ' to be at most 20% higher than θ, the result of the first optimization step. The absolute traffic variables x_{ij}^k are replaced with relative traffic variables α_{ij}^k. As a result, equations 7 and 8 have almost the same syntactic formulation and exactly the same semantic function as equations 4 and 5 in Problem 2.

The traffic variables are changed from absolute to relative values to enable an efficient introduction of two new sets of

$$\min \quad \theta' \frac{1}{2\,\lambda\,\theta} + \sum_{ijk} u_{ij}^k - \sum_{ij} v_{ij}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_k \alpha_{ij}^k = 1 && \forall ij && (7)\\
& \sum_{ij} \sum_k g_{ij}^k(e)\,\alpha_{ij}^k\,t_{ij} \leq \theta'\,c(e) && \forall e && (8)\\
& \theta' \leq \lambda\theta && && (9)\\
& \alpha_{ij}^k \leq u_{ij}^k && \forall ijk && (10)\\
& \alpha_{ij}^j \geq v_{ij} && \forall ij && (11)\\
& \alpha_{ij}^k \geq 0 && \forall ijk && (12)\\
& u_{ij}^k, v_{ij} \in \{0,1\} && \forall ijk && (13)
\end{aligned}$$

Problem 3: Tunnel Minimization Extension (TME). MILP as second optimization step to Problem 2 that minimizes the number of SR tunnels.

$$\min \quad \theta' \frac{1}{2\,\lambda\,\theta} + \sum_{k \neq j} u_{ij}^k$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_k u_{ij}^k = 1 && \forall ij && (14)\\
& \sum_{ij} \sum_k g_{ij}^k(e)\,u_{ij}^k\,t_{ij} \leq \theta'\,c(e) && \forall e && (15)\\
& \theta' \leq \lambda\theta && && (16)\\
& u_{ij}^k \in \{0,1\} && \forall ijk && (17)
\end{aligned}$$

Problem 4: Tunnel Limit Extension (TLE). MILP as second optimization step to Problem 2 that minimizes the number of SR tunnels and limits the number of tunnels per demand to 1.

binary variables u_{ij}^k and v_{ij}. As a result of introducing binary variables, the problem is now a Mixed Integer Linear Program (MILP). The first set of new variables u_{ij}^k provide the ability to count the total number of traffic variables used. They are set to 1 if and only if the corresponding α_{ij}^k is larger than 0 (equation 10). Minimizing the total number of SR tunnels is of little use, because tunnels that follow the shortest path of their demand do not have to be installed in practice. Shortest path tunnels should, therefore, be counted and weighted in a different way than actual tunnels.

Proposition 1: A demand with ingress node i and egress node j is routed along the shortest path in 2-Segment Routing iff $\alpha_{ij}^k = 1$ with $k = j$.

The second set of variables v_{ij} is used to count these cases. They are set to 1, iff the corresponding α_{ij}^j is exactly 1 (equation 11). Then, the objective can be defined as a minimization of the sum of all variables u_{ij}^k minus the sum of all shortest path tunnels v_{ij}. The maximum link utilization θ' is added to the objective function to rate solutions with an equal number of tunnels. The constant coefficient ensures that the impact of θ' will always be lower than the impact of a single binary variable.

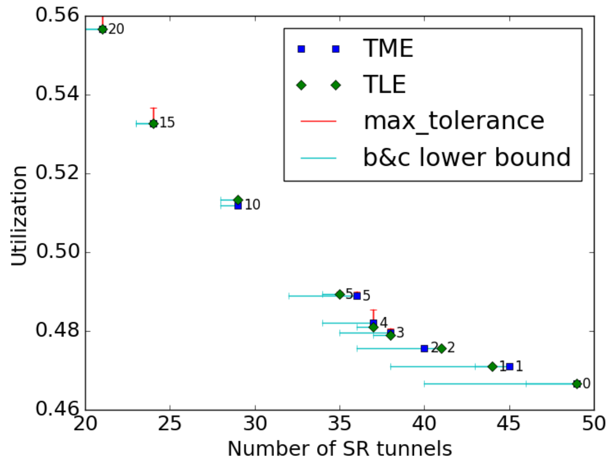

Figure 5: Results of 2-SR extensions for different tolerance levels for the instance with the highest number of tunnels.

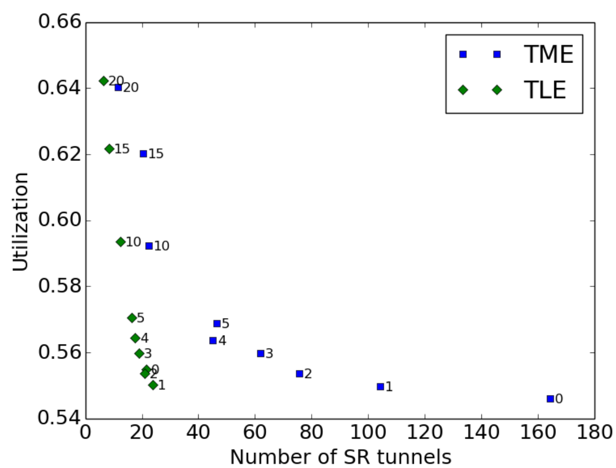

Figure 6: Average results of 2-SR extensions through all 16 instances. TME results were adjusted to show the exact number of tunnels when taking split-factors into account.

B. Tunnel Limit Extension (TLE)

In this form, the Tunnel Minimization Extension allows multiple tunnels for each demand, like in the example in Figure 1. This may not be wanted due to complexity or hard constraints in deployment that appear when trying to introduce multiple tunnels with arbitrary weights. In this case, the program can be simplified to only allow one tunnel per demand. The simplified MILP formulation for the Tunnel Limit Extension is shown in Problem 4.

When limiting the number of tunnels per demand to 1, the meaning of α_{ij}^k and u_{ij}^k become the same. Dropping the α variables, the u variables are kept for consistency, as their binary nature is still needed for the formulation to work. Apart from this change, the constraints stay the same. Only the objective needs an additional change, as the shortest path tunnels should not be counted. To do this, the sum simply skips all u_{ij}^k variables where $k = j$.

C. Evaluation

We evaluated the TME and TLE using our reference scenario set. Due to the increased complexity in solving a MILP, a time limit of three hours per instance was set. Thus, the results are not optimal in all cases, but only near optimal. The outcomes of TME as well as TLE are shown in Figure 5. For simplicity, we only display the instance with the highest number of tunnels. Different values of λ are shown in percentages from 0% to 20%. Each point is defined by the number of tunnels on the x axis and the maximum link utilization on the y axis. In addition, each point has a vertical red line, which shows the distance to the maximum allowed utilization specified by $\lambda\,\theta$. The horizontal blue line displays the distance to the theoretical lower bound at the time the second optimization phase was terminated due to the time limit. Both lines together give a good impression of theoretical optimal results.

We see that the number of tunnels are significantly reduced. With either of the two extensions, the number of tunnels is reduced to two digit numbers. Due to the additional limitation when including equation 12, TLE shows to be infeasible at zero percent tolerance for some of the instances, but not in the worst case instance. For most other tolerance levels, TLE requires as many or less tunnels than TME. This is not an expected behaviour, as TLE is much more restricted than TME. While TME is allowed to split the demands, only in few cases two or at most three tunnels per demand are used. Also, TME uses much more computation time and memory to get to reasonable results. This can be attributed to the higher complexity of the TME formulation, while the additional restrictions for TLE lead to a smaller and simpler formulation. The argument is supported by the blue horizontal lines, which are almost exclusively present for TME. They show that there could be a potential to get better results than TLE when given even more time and memory. It is notable, though, that TME is able to give a better utilization for the 2% tolerance level, which supports this point.

Altogether, the results show that the number of tunnels can be reduced to a reasonable scale, even on low tolerance levels.

Apart from increased running time and memory consumption, there is another major problem when using TME. As indicated in section III-B, arbitrary split-factors can not be directly applied to a real network due to hardware restrictions. Thus, the tunnel values given by TME have to be reconsidered, taking into account the respective split-factors. For one traffic demand, the optimized split-factors are mapped and discretized to parts of 32, as this is a common maximum split-value for routers. Then, all factors are viewed as fractions and reduced to the smallest denominator. The highest denominator of all factors of one demand then dictates the number of tunnels that are actually needed for the demand. For example, if we consider a demand with three tunnels and split-factors 0.24, 0.26 and 0.5, this would map to the fractions $8/32$ for the first two values and $16/32$ for the third. Through reduction we get $1/4$ and $1/2$. The highest denominator is 4, so to deploy our example, actually 4 tunnels are required instead of the promised 3.

The adjusted results are plotted in Figure 6. Intuitively, this additional computation that the optimization wasn't aware of dramatically increases the numbers for TME. To have a smoothed view, the average value over all 16 instances are shown. The point of 0% tolerance has to be ignored for TLE, because some of the instances were infeasible at that level and, thus, we have a smaller set of instances to draw the average from. The recomputed TME now requires three to eight times more tunnels than TLE at low tolerance levels. As a result, TLE looks even stronger as it did before.

VII. Summary

Thanks to the data from a European tier one ISP, an analysis about traffic demands was possible. We showed that the traffic peak is around a quarter to ten both on weekdays and on the weekend. Also, American and Asian traffic have only a marginal influence on Europe and the differences in time zones have no impact on the choice of representative matrices. Besides, the seemingly natural assumption that Internet traffic increased over the last few years was confirmed. Directly following this potentially problematic point, it was noticed that the network was expanded well to counteract the growth in the past.

We implemented three algorithms to evaluate TE with real backbone traffic. Choosing arbitrary paths with MCF has shown a lower bound for network utilization. A simple shortest path routing was used to determine the current state of the art. As MCF is difficult to deploy in practice, a SR variation was implemented as the third algorithm to optimize network utilization. It was shown that the performance of defining only one intermediate node already leads to nearly the same results as MCF. This validates the findings in [3], now using real-world traffic and topologies. In addition, two extensions were introduced. The goal of both extensions was to minimize the number of non-shortest path tunnels. The second variant additionally limited the total number of tunnels per demand to one. The overall goal to make 2-SR ready for deployment from a network operators perspective was shown to be reached. The best results were achieved using TLE with low tolerance parameters. It does not seem profitable to allow arbitrary splitting of demands, as TME is far more complex and ressource intensive without providing any improvements in result quality.

In the future, we plan to continue working on additional real-world constraints. In order to keep the number of SR tunnels low, the maximum link utilization had to be penalized. The question whether a 3 or 4-SR approach would lead to a smaller penalty in order to minimize the number of tunnels and, therefore, better results is an open question. Furthermore, we plan to investigate whether the tunnel extension algorithms return similar tunnels for the same topology but different traffic matrices. If so, it would be interesting to actually deploy some of the more important tunnels to see their stability when provided with previously unknown traffic demands.

Acknowledgements

This work was supported in part by the "Stifterverband für die Deutsche Wissenschaft" (H170 5701 5020 20951).

References

[1] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic Engineering in Software Defined Networks," in *Proc. of the 2013 IEEE Conf. on Computer Communications (INFOCOM)*, 2013, pp. 2211–2219.

[2] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to SVP for LSP Tunnels," RFC 3209, 2001.

[3] R. Bhatia, F. Hao, M. Kodialam, and T. V. Lakshman, "Optimized Network Traffic Engineering using Segment Routing," in *Proc. of the 2015 IEEE Conf. on Computer Communications (INFOCOM)*, 2015, pp. 657–665.

[4] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2015–2020," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf, 2016.

[5] K. Claffy, G. Miller, and K. Thompson, "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone," in *Proc. of the 1998 INET Conf.*, vol. 98, 1998, pp. 21–24.

[6] K. G. Coffman and A. M. Odlyzko, "Internet Growth: Is There a "Moore's Law" for Data Traffic?" in *Handbook of Massive Data Sets*, ser. Massive Computing, J. Abello, P. M. Pardalos, and M. G. C. Resende, Eds. Springer US, 2002, vol. 4, pp. 47–93.

[7] R. J. Essiambre and R. W. Tkach, "Capacity Trends and Limits of Optical Communication Networks," *Proc. of the IEEE*, vol. 100, pp. 1035–1055, 2012.

[8] S. Even, A. Itai, and A. Shamir, "On the Complexity of Time Table and Multi-Commodity Flow Problems," in *Proc. of the 1975 Annual Symp. on Foundations of Computer Science (FOCS)*, 1975, pp. 184–193.

[9] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The Segment Routing Architecture," in *Proc. of the 2015 IEEE Global Communications Conf. (GLOBECOM)*, 2015, pp. 1–6.

[10] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World," *IEEE Journal on Selected Areas in Communitacions*, vol. 20, pp. 756–767, 2002.

[11] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," RFC 7426, 2015.

[12] R. Hartert, P. Schaus, S. Vissicchio, and O. Bonaventure, "Solving Segment Routing Problems with Hybrid Constraint Programming Techniques," in *Proc. of the 2015 International Conf. on Principles and Practice of Constraint Programming (CP)*, 2015, pp. 592–608.

[13] IBM, "IBM ILOG CPLEX Optimization Studio V12.6.0," http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.0, 2014.

[14] Juniper, "Junos OS Administration Library for Routing Devices: maximum-ecmp," http://www.juniper.net/documentation/en_US/junos15.1/topics/reference/configuration-statement/maximum-ecmp-edit-chassis.html, 2013.

[15] D. Medhi, *Network routing: algorithms, protocols, and architectures*. Morgan Kaufmann, 2010.

[16] D. L. Mills, "Internet Delay Experiments," RFC 889, 1983.

[17] J. Moy, "OSPF Version 2," RFC 2328, 1998.

[18] D. Oran, "OSI IS-IS Intra-domain Routing Protocol," RFC 1142, 1990.

[19] B. Rogan and I. Swett, "QUIC, Next generation multiplexed transport over UDP," https://www.nanog.org/sites/default/files//meetings/NANOG64/1051/20150603_Rogan_Quic_Next_Generation_v1.pdf, 2015.

[20] E. Rosen, A.Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, 2001.

[21] S. Schnitter and M. Horneffer, "Traffic Matrices for MPLS Networks with LDP Traffic Statistics," in *Proc. of the 2004 Telecommunications Network Strategy and Planing Symp. (NETWORKS)*, vol. 11, 2004, pp. 231–236.

[22] S. Sen and J. Wang, "Analyzing Peer-To-Peer Traffic Across Large Networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 12, pp. 219–232, 2004.

[23] J. Van der Merwe, S. Sen, and C. Kalmanek, "Streaming Video Traffic: Characterization and Network Impact," in *Proc. of the 2002 International Web Content Caching and Content Distribution Workshop (WCW)*, 2002.