

Key words: traffic engineering, network provisioning, network design

BINGJIE FU, STEVE UHLIG*

ON THE RELEVANCE OF ON-LINE TRAFFIC ENGINEERING

The evaluation of dynamic Traffic Engineering (TE) algorithms is usually carried out using some specific network(s), traffic pattern(s) and traffic engineering objective(s). As the behavior of a TE algorithm is a consequence of the interactions between the network, the traffic demand and the algorithm itself, the relevance of TE may depend on several network design aspects.

In this paper, we evaluate well-known TE algorithms using real-world and generated network topologies and traffic demands. By re-scaling observed traffic demands, we are able to observe the behavior of TE algorithms under a variety of situations, which may not be observable in reality. We identify distinct network load regimes that correspond to different behaviors of the TE algorithms. We also study the impact of several network design aspects, like network provisioning and redundancy, on the relevance of TE algorithms. We find that there are specific situations under which TE algorithms are useful. These situations depend highly on shortcomings in the network provisioning as well as on the availability of alternative paths in the network.

1. INTRODUCTION

The evolution of the Internet has been amazing in the last decades. New applications with different service constraints have emerged, such as real-time video conference, on-line gaming, etc. High-speed networks are expected to support a wide variety of these sensitive applications. However, the current Internet architecture offers mainly a best-effort service, so providing Quality of Service (QoS) guarantees in the current Internet is not easy. More demanding applications as well as stricter SLA's are driving Internet Service Providers to rely on traffic engineering [1] to better control the flow of IP packets.

In pure IP networks, traffic engineering is implemented by changing the intradomain routing protocol (called Interior Gateway Protocol or IGP) weights [9]. Optimizing the IGP weights to force traffic to follow high-capacity and/or low-delay paths is a way to perform traffic engineering. When coupled with on-line TE algorithms, the label-based forwarding mechanisms such as Multi Protocol Label Switching (MPLS) [20] and Generalized Multi Protocol Label Switching (GMPLS) [14] make per-flow path selection with service guarantees possible. The label-based forwarding mechanisms provide an opportunity to control traffic in a finer-grained way than by changing IGP weights.

Traffic engineering requires the computation of paths between each pair of routers. This computation may be done off-line or on-line. The computation is done off-line when paths do not need

*Delft University of Technology, {B.Fu,S.P.W.G.Uhlig}@ewi.tudelft.nl

to adapt to traffic dynamics. On-line computation, on the other hand, is required when the routing plans must adapt to changing network conditions. On-line TE techniques leverage the capability of label-based forwarding mechanisms to choose paths inside the network to better utilize the available resources in the network. On-line TE algorithms select for each new connection request a path with enough resources, based on the current state of the network. In this paper, we focus exclusively on on-line traffic engineering algorithms.

On-line TE algorithms need to address two different issues:

- Find a routing path through the network with enough resources (e.g. bandwidth), called a *feasible path*;
- Try to accommodate as many requests as possible.

To use the network capacity in the most efficient way, the routing paths chosen for the traffic should not interfere with each other too much [12]. In practice, preventing interference between requests is difficult due to the low number of links in a network and the limited resources available.

Many factors affect the outcome of the path finding and resource allocation process by on-line TE algorithms. The topology structure, the link costs, the residual link capacities, the path selection, etc, affect the global performance of the TE algorithms. Obtaining a priori knowledge of the behavior of a TE algorithm on a particular network scenario is very hard due to the interactions between the different aspects of the problem.

TE algorithms are usually designed to optimize a particular objective function and address a specific traffic engineering problem [10, 12, 2, 24]. Setting up scenarios under which TE algorithms will be compared is difficult because some TE algorithms have not been designed to perform well under specific situations. Not all scenarios are practically relevant. If some TE algorithm performs poorly under a scenario that is not relevant in practice, this should not undermine the suitability of the TE algorithm.

In this paper we categorize the different factors that affect the behavior of TE algorithms as follows:

- network topology: The location of Points of Presence (POP) and the existing connectivity (e.g. fiber) between POPs largely determine the network structure. Most ISPs design their networks with specific redundancy requirements, leading to rather sparse topologies that trade-off cost and robustness [3].
- traffic demand: The amount of traffic between each pair of routers.
- network provisioning: Network provisioning consists in scaling the link capacities, so as to accommodate the traffic between all pairs of routers.

The network, traffic and routing are intertwined. The design of the network (capacities and link weights) must ensure that the traffic demand can be satisfied. To assign link capacities, network design requires an initial model of routing to determine how much traffic will flow on each link. In today's networks, link weights are set based on the delay, the capacity [18], or a combination of delay and capacity of the links. The goal of setting IGP weights in such a way is to attract traffic towards high-capacity and low-delay links. In practice, many backbone operators use the ad-hoc approach of observing the flow of traffic through the network [7], and iteratively adjusting a weight whenever the load on the corresponding link is higher or lower than desired. The problem has been addressed in [10, 9, 17] using different techniques from operations research. Observing traffic in large backbone networks requires significant resources [7], so that in practice the real traffic demand is inferred rather than observed [15, 25]. Few observed traffic demands are publicly available [24, 23].

In this paper, we study the behavior of on-line TE algorithms, using a set of real and synthetic traffic demands and backbone network topologies. To our knowledge, this is the first time that the relationship between network design and the behavior of on-line TE algorithms is studied in such a way. Our contributions consist in identifying several factors that determine the relevance of on-line TE algorithms. Due to the small viability of different time scales [22], in this paper we use arbitrary scalings, which are supposed to show the general behaviors.

The remainder of this paper is organized as follows. We review the literature by presenting some well-known TE algorithms in Section 2. The behavior of TE algorithms on two real-world scenarios is studied in Section 3. In Section 4, we build our own scenarios to study the impact of network design aspects on TE algorithms.

2. RELATED WORK

Many Autonomous Systems (ASes) use Open Shortest Path First (OSPF) [16] or Intermediate System-Intermediate System (IS-IS) [4] as their intradomain routing protocol. These protocols select shortest paths based on static link weights. These IGP link costs reflect the desirability of a link to be selected to carry traffic inside the AS. We call the routing algorithm used by these routing protocols the *Static Shortest Path* algorithm (SSP), as the shortest paths will remain the same no matter the amount of traffic actually carried on those links, as long as the link weights do not change. The link costs can be computed off-line according to the offered traffic, the provisioned network capacity, and the specific objective desired by the network administrator [10], or simply by following Cisco's guideline of IGP costs proportional to the inverse of the link capacity [5]. Recently, splitting traffic among several shortest-path, called equal-cost multi-path (ECMP) [16], has been deployed by an increasing number of network operators. ECMP provides some load-balancing and makes the granularity of the traffic finer compared to non-ECMP.

The algorithms that use the off-line computed link costs, are called off-line routing algorithms. SSP is computationally efficient, but unaware of the link utilization. If the link costs setting is far from the optimal setting with respect to the current traffic demand, then SSP may heavily load some links while avoiding others that have plenty of unused capacity. When capacity is not available on the shortest path, IP packets are simply dropped as the router buffers saturate or connection requests are rejected as feasible paths do not exist.

On-line routing algorithms have been proposed to find paths with bandwidth requirements. On-line algorithms use dynamic information such as the link residual capacities to compute the feasible paths, and are able to find one, provided it exists. There is an extensive literature on on-line TE algorithms. Due to space limitations, we do not discuss all of them. In this paper, we select a set of representative TE algorithms, from simple to computationally complex ones.

The Widest Shortest Path algorithm (WSP) [11] computes the shortest IGP paths in the network formed by links with sufficient residual capacities to accommodate the incoming request, and selects from all the feasible shortest IGP paths the one with the maximum bottleneck residual capacity to load-balance network traffic. It avoids using heavily loaded links unless there is no other option. Only feasible shortest IGP paths are considered.

Another instance of on-line algorithm is to use shortest-path routing with dynamic link weights. Instead of using SSP with a static inverse of the link capacity as the link weights, one can use the inverse of the residual capacity. As links get loaded, the weights will adapt to reflect the available capacity in the network. We call this technique dynamic shortest path with inverse residual capacity

(DSP-Inv).

The family of Minimum Interference Routing Algorithms (MIRA) [12] tries to select, between each source-destination pair, a path that interferes the least with other source-destination pairs. MIRA takes into account the information of source-destination pairs (S_i, D_i) and weights them by their importance α_i . The importance of a source-destination pair can be set for example as the fraction of the total traffic it represents. To minimize interference, those algorithms maximize the sum of the residual weighted max-flows[†] between all source-destination pairs. Upon the arrival of a connection request from S_i to D_i , the algorithm will select a path that can maximize

$$\sum_{(S_j, D_j), j \neq i} \alpha_j \theta_j \quad (1)$$

where θ_j represents the residual max-flow for (S_j, D_j) after a path is selected for (S_i, D_i) . Maximizing this objective function is NP-hard [12]. The MIRA algorithms reach a suboptimal solution by computing the set of *critical* links L_j for all other pairs (S_j, D_j) ($j \neq i$). A link l is called *critical* for a source-destination pair (S, D) , if the reduction in l 's capacity leads to the reduction in (S, D) 's residual max-flow. Then link costs are set according to the link *criticality*, and shortest path computation is applied to this weighted topology. When only the importance of each source-destination pair is considered in the link *criticality*, the link cost can be set as

$$C(l) = \sum_{(S_j, D_j), l \in L_j} \alpha_j \quad (2)$$

or as

$$C(l) = \sum_{(S_j, D_j), l \in L_j} \alpha_j / R(l) \quad (3)$$

when the residual capacity of each link is also considered, where $R(l)$ is the residual capacity of l . When MIRA is using equation 2 (equation 3) for its link costs, it is called MIRA-TM (MIRA-TM-Cap). For further details about MIRA we refer to [12].

3. BEHAVIOR OF TE ALGORITHMS ON REAL-WORLD NETWORKS

In this section, we observe the behavior of on-line TE algorithms on real-world networks and real traffic demands. In Section 4 we rely on generated topologies to show the impact of different aspects of network design on the behavior of TE algorithms.

3.1. SCENARIOS

We use real topologies and traffic matrices from Abilene[‡] and GEANT[§]. Abilene and GEANT are the US and European academic backbones. They both provide transit service to universities and research institutions in the US and Europe. The Abilene topology has 12 nodes and 30 directed links while GEANT has 22 nodes and 72 directed links. The link capacities and link costs are kept as in reality, with link costs in Abilene assigned based on the geographical distance, and that in GEANT

[†]The max-flow of a source-destination pair (S, D) is the maximum amount of traffic that can be pushed between this pair. Multiple link-disjoint paths may be used.

[‡]<http://abilene.internet2.edu/>

[§]<http://www.geant.net/>

following a modified version of Cisco’s proposal (proportional to the inverse of the link capacity). For Abilene [24], a traffic matrix spans 5 minutes, and for GEANT [23] 15 minutes. When using a traffic matrix, for each source-destination pair, we deduce the average bandwidth requirements from the traffic matrix. The traffic between each source-destination pair is further split into small pieces to create the requests for path establishment inside the network. The traffic demand from the traffic matrix between each source-destination pair is split into 200 equal pieces. In practice, the order in which the connection requests arrive may have an impact on the behavior of the TE algorithm. If the order of the connection requests is important for the performance of the TE algorithm, then an off-line TE algorithm should be used in order to properly schedule the requests. As we are interested in on-line TE algorithms in this paper, we do not consider the interactions between the connection requests. Therefore, the requests we use are small enough to be close to a fluid for the network[¶]. Connection requests are handled in the following way:

- Select uniformly an source-destination pair from all potential ones;
- Inject a connection request between this pair with a corresponding bandwidth requirement;
- Reserve the capacity for this connection request if a feasible path is found by the algorithm; the capacity will reserved for this connection until the end of each analysis.

No matter which algorithm is used to compute the paths (including SSP), a reservation is made in the network if a feasible path is found by this algorithm.

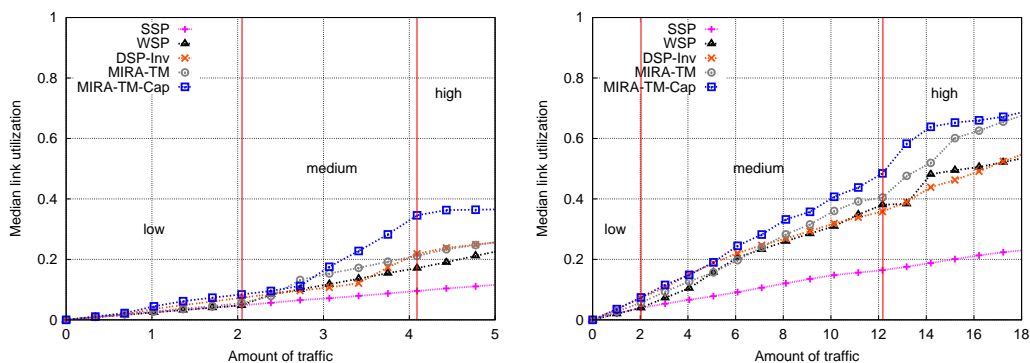


Figure 1: Median link utilization under different load regimes: Abilene (left) and GEANT (right).

By adjusting the total number of injected connection requests, different *load regimes* can be reached. We show in Section 3.2 how different *load regimes* appear when scaling traffic demands.

To check our results with the literature, we also borrowed the scenario from [12]. In this topology, each link is bidirectional (can be treated as 2 independent unidirectional links with the same capacity). Some links have capacities of 1200 units and the others have 4800 units. The connection requests arrive randomly, with the same rate for all given source-destination pairs, and with a bandwidth requirement uniformly distributed between 1 and 3 units. The connection requests are injected into the network in the same way as explained for Abilene and GEANT.

3.2. IMPACT OF TRAFFIC LOAD

In this section we provide simulation results of the different TE algorithms applied to both Abilene and GEANT. For each network, we pick one traffic matrix out of several measured ones

[¶]We considered coarser connection requests and noticed that increasing their size does not significantly change our results, but makes the curves appear less smooth.

during peak time. Figure 1 shows the median link utilization after each connection request is routed by the TE algorithms. The median link utilization is the value such that half of the links in the network have a smaller utilization. We use the median because we want to have a picture of the global link utilization in the network, and because it is less sensitive to extreme values than the average. The average link utilization provides similar results. The wide variations between the minimum and the maximum link utilization make those statistics not insightful.

To see how TE algorithms manage to handle different traffic loads, we inject requests proportionately to the real traffic matrices until the connection requests for a significant fraction of the source-destination pairs cannot be accepted. By doing this, we ensure that we sample all *load regimes* that correspond to the traffic distribution given by the traffic matrix.

The results for the following TE algorithms are shown on Figure 1: SSP, WSP, DSP-Inv, and 2 variants of MIRA (MIRA-TM and MIRA-TM-Cap). MIRA-TM considers in its weighting the traffic between source-destination pairs (see equation 2). MIRA-TM-Cap takes into account not only the traffic between source-destination pairs, but also the residual capacity of the critical links (see equation 3).

The x-axis of both graphs of Figure 1 give the sum of the connection requests that have been pushed into the network so far (total traffic that has been accepted). We scaled this sum to quantify the traffic demands in terms of the measured traffic matrices, without revealing the exact amount of traffic that a single traffic matrix represents. The left graph in Figure 1 makes three different *load regimes* appear: low load, medium load, and high load. When amount of traffic is low enough, the algorithms can be divided into two groups: 1) SSP and WSP; 2) DSP-Inv and the MIRA family. SSP and WSP share the same link costs settings which do not give multiple shortest IGP paths in the Abilene network, so when the load is low enough SSP and WSP will choose the same paths. DSP-Inv and the MIRA algorithms on the other hand do not follow the static link costs settings of the network, but use longer paths with respect to the static IGP costs to avoid critical links and hence reduce interference with other requests. If network administrators prefer low link utilization, TE algorithms like DSP-Inv or MIRA-TM-Cap perform worse than SSP and WSP as seen from the graph for median link utilization.

In what we call the medium load regime, SSP and WSP choose different paths. WSP chooses longer paths on average, like MIRA, hence the median link utilization increases faster than for SSP. When the network starts to become loaded, WSP detects that some shortest paths are not feasible any more and thus turns to alternative paths. SSP always uses the paths computed based on the static link costs settings, resulting in blocked connection requests when load becomes too high. One of the interests of TE algorithms lies in their ability to find feasible paths as long as there exists some, even when load is becoming high in the network.

Finally, the high load regime occurs when not enough capacity is left in the network. In this situation, even the MIRA family is unable to satisfy connection requests between a significant fraction of the source-destination pairs. This situation should never happen in practice, as it indicates inadequate network provisioning.

For the GEANT network (right graph Figure 1), we also observe three distinct load regimes. In the low load region SSP and WSP behave in the same way. Then in the medium load region WSP chooses different paths from those chosen by SSP and has a higher median link utilization. Finally, in the high load regime, the median link utilization hardly increases due to lack of available capacity to satisfy connection requests.

3.3. AVAILABLE CAPACITY IN THE NETWORK

Figure 1 showed that the median link utilization stays relatively low, even under the high load regime. Although many links still have capacity left, TE algorithms are unable to satisfy a significant fraction of the connection requests. As paths between source-destinations are typically made of several links, connection requests between most of the source-destination pairs cannot be satisfied, as soon as a significant fraction of the links are highly loaded.

Link utilization-related metrics give a picture that is easy to understand for network operators, as it is closely related to delay [19]. To understand the behavior of TE algorithms on the other hand, a metric based on the capacity left in the network is more helpful. For this, we rely on the sum of the residual max-flows of all source-destination pairs. The residual max-flow of a particular source-destination pair tells how much usable capacity is left in the network to satisfy connection requests between the considered source-destination pair. The sum of the residual max-flows for all source-destination pairs tells how much capacity is usable by TE algorithms to satisfy connection requests.

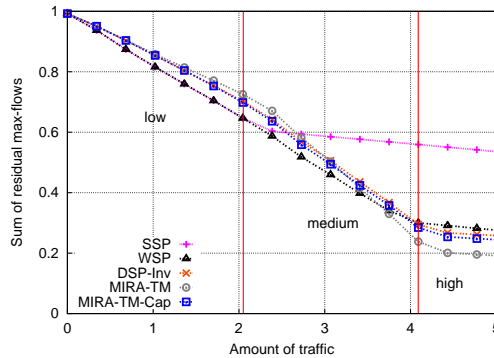


Figure 2: Network-wide residual max-flow on Abilene network.

Figure 2 shows how the sum of residual max-flows for all source-destination pairs evolves for increasing load in the Abilene network (x-axis is the same as on Figure 1). Note that the value of the sum of the residual max-flows is meaningless, only the relative speed at which it decreases for different TE algorithms is of relevance. We thus normalize it in all figures.

In the low load regime, there is no big difference between the algorithms. SSP and WSP use a bit more of the total max-flow than the other three algorithms. In the medium load regime, SSP cannot satisfy some connection requests, hence does not use the available max-flow. WSP, DSP-Inv and the two MIRA variants manage to explore available paths to route connection requests. In the high load regime, the four dynamic algorithms hardly manage to satisfy connection requests. Hence, they have a very slowly decreasing residual max-flow.

Overall, on-line TE algorithms behave in a similar way. They use longer paths than SSP to make a better use of the available capacity. Unless those longer paths do not interfere with later requests from other source-destination pairs, TE algorithms should perform equally well on the scale of the whole network, as measured by the sum of the residual max-flow. As networks contain less links than source-destination pairs, it is unlikely that alternative paths that do not interfere with other source-destination pairs may be found in the network.

To better understand why TE algorithms behave globally in a similar way, we need to focus on specific source-destination pairs. To make the explanation simpler, we take the same topology as used in [12]. The MIRA algorithms were proposed to try to minimize interference between different source-destination pairs. The insight behind MIRA is that in order to better utilize the available

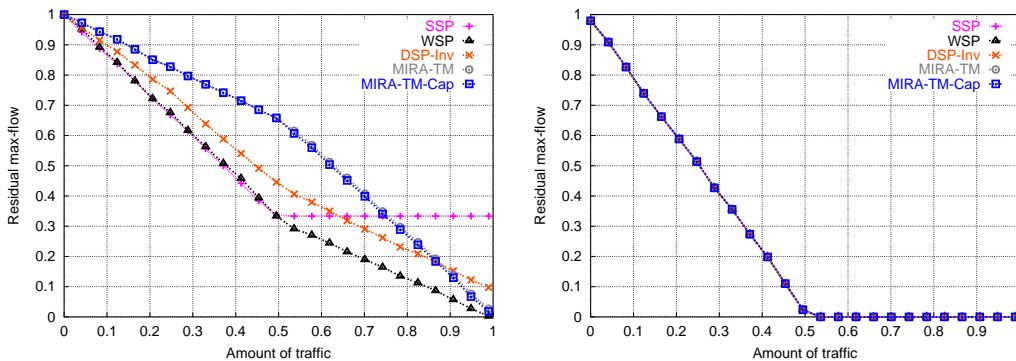


Figure 3: Residual max-flow for pair (S_1, D_1) (left) and pair (S_2, D_2) (right)

capacity in the network, each source-destination pair should try to prevent interfering with the residual max-flow of other source-destination pairs when choosing their path. Another important aspect of interference is that different source-destination pairs have different possibilities to have their paths routed in order to minimize interference. We implemented the scenario of [12] and computed after each request the residual max-flow of each source-destination pair.

Figure 3 shows the residual max-flows of 2 of the 4 source-destination pairs used in [12], (S_1, D_1) and (S_2, D_2) . For (S_1, D_1) , we obtain results similar to those shown in [12]. The two variants of the MIRA algorithm leave a larger residual max-flow compared to the other three algorithms for an amount of traffic smaller than 0.75. Next comes DSP-Inv, and finally WSP and SSP. At an amount of traffic of 0.5, SSP cannot use the whole max-flow of (S_1, D_1) due to its inability to choose alternative paths. DSP-Inv is also unable to use the whole max-flow of (S_1, D_1) , although to a smaller extent than SSP. WSP and the two MIRA variants on the other hand manage to use the whole max-flow of (S_1, D_1) .

For (S_2, D_2) (bottom of Figure 3), the residual max-flows found for all the considered algorithms are the same. Although there are two bottleneck links forming (S_2, D_2) 's max-flow, both links are also used by (S_4, D_4) 's max-flow. No TE algorithm can prevent interference to happen for (S_2, D_2) .

From a network-wide perspective, TE algorithms alleviate problems on some links by shifting it to other links. How much TE helps, however, depends much on the considered source-destination pair. When alternative paths can be used without interfering with other source-destination pairs, TE algorithms may help. When critical links cannot be bypassed, TE cannot compensate for inadequate provisioning in the network or for links that belong to the residual max-flow of several source-destination pairs.

3.4. IMPACT OF TRAFFIC PATTERN

Network traffic is dynamic, it exhibits daily and weekly patterns [6, 21, 8]. Different traffic patterns might complexify the picture of TE algorithms we have shown so far. In the previous section, we showed the impact of the load regime on the behavior of TE algorithms. Over time, both the total amount of traffic and the distribution of the traffic among source-destination pairs may change. On Figure 4, we show the distribution of the traffic among source-destination pairs for four different traffic matrices of Abilene. We selected two days, June 1st and 2nd 2004, and within those two days we selected two time intervals, one during peak time (16:55) and another during non-peak time (21:35).

We observe on Figure 4 that the distribution of the traffic differs much between peak and non-

peak time for Abilene. During peak time, one particular source-destination pair is responsible for about 60% of the total traffic. During non-peak time on the other hand, the traffic distribution among source-destination pairs is less uneven, but still far from uniform. During non-peak time, 20 among the 132 source-destination pairs are responsible for about 50% of the total traffic. Note that the variations in the traffic demand of GEANT are limited and do not impact our results.

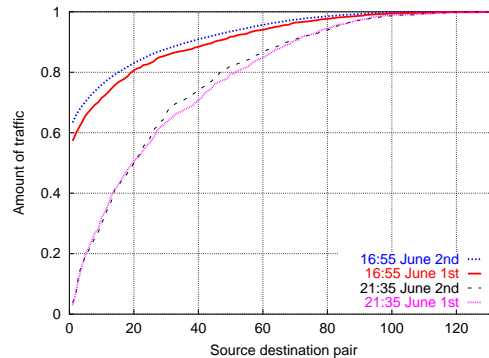


Figure 4: Distribution of traffic among source destination pairs (Abilene).

Similarly to Section 3.2 and 3.3 where the results under traffic matrix of Abilene taken at 16:55 June 1st were shown, we performed the comparison of TE algorithms on several other traffic matrices (the other three instances whose distributions are shown in Figure 4) of Abilene to see how different traffic matrices affect the comparison. Due to the space limitation, we do not show the results for these traffic matrices in this paper. Surprisingly, we do not observe major differences with the results from Section 3.2 and 3.3. As soon as the network becomes loaded, we observe different behaviors of the TE algorithms, corresponding to the medium or high load regimes. When the amount of traffic is low, then the TE algorithms behave as in the low load regime.

While traffic matrices for 16:55 June 1st and 16:55 June 2nd push the network to experience the 3 load regimes with traffic scale 5, the figures for 21:35 June 2nd show a linear increase in median link utilization and decrease in residual max-flow. The reason is that the traffic matrix taken at 21:35 corresponds to relatively low traffic, compared with the ones taken at 16:55 which correspond to peak time.

3.5. TE ALGORITHMS AND LOAD REGIMES

From the results of this section, we identified one main factor that affects the behavior of the TE algorithms: the total amount of traffic the network has to handle. This total amount of traffic translates, through the choice of the paths made by the routing algorithms, into a load regime. We are now in a position where we can loosely define the three load regimes we observed in this section. The low load regime is defined by a behavior of the TE algorithms that is similar to the one of SSP. In this situation, it is not necessary to use non-shortest paths to accept the connection requests. As soon as SSP is not able to satisfy some connection requests, we enter the "medium load regime". In this medium load regime, TE algorithms manage to better use the available capacity than SSP, to accept connection requests that would be blocked otherwise. Finally, when the amount of traffic is so high, that even TE algorithms are unable to satisfy connection requests, we enter the "high load regime". In this high load regime, the solution is to increase the capacity of the links or to add redundant links.

4. IMPORTANCE OF NETWORK DESIGN ASPECTS

Section 3 showed the differences between TE algorithms under different load regimes. As we relied on real-world networks and traffic demands, we could not pinpoint the importance of specific aspects of network design on the behavior TE algorithms. In this section, we build up scenarios to study the impact of the network topology and network dimensioning. We propose scenarios based on differently connected topologies, with different ways of assigning link capacities and generating traffic demands.

Topology Our topologies are generated using the iGen topology generator^{||}. iGen allows to generate random points in one or any continent, and then to connect the nodes using network design heuristics [3]. It can also set capacities of the links and IGP link weights (based on physical distance or the inverse of the link capacity). We choose topologies with 25 nodes, and randomly generated points in Northern-America.

Link weights The weight for each link l_{ij} is assigned as a piecewise linear function of the geographical distance between node i and j .

Traffic demand and capacity provisioning For each topology, two combinations of traffic demand and capacity provisioning are considered:

- *properly provisioned networks*: We generate the traffic demand based on a gravity model [13]. First, we generate for each edge node i (source or destination) a total traffic demand x_i following a uniform distribution. The amount of traffic X_{ij} from node i to node j is set proportional to the product of the traffic demands x_i and x_j , i.e., $X_{ij} = \beta x_i x_j$, where β is some constant. We assign capacities to links by first assigning paths to source-destination pairs. Paths are obtained by running a shortest path using the already assigned link weights. Once a link is used by the path from node i to j , the capacity of this link will be increased by 1.1 times the corresponding traffic amount X_{ij} . The factor 1.1 provides a very small over-provisioning. When links are not used by any shortest path, they would end up with 0 capacity. We assign to these links the average capacity of the non-empty links.
- *Improperly provisioned networks*: The traffic and link capacities are considered separately. All links are assigned the same capacity. All source-destination pairs have the same total amount of traffic, but connection requests have a uniform size between 1 and 3 units of traffic.

In the *properly provisioned networks*, the capacities of links are assigned in such a way that the provisioning of the network matches the traffic matrix, under the assumption that shortest paths are used. In *improperly provisioned networks*, the network capacities are not designed to match the traffic demand at all. We thus expect different behaviors of the TE algorithms under the two types of networks.

4.1. MINIMALLY-CONNECTED TOPOLOGY

We start by relying on the worst possible connectivity a network can have: a tree. We use the MST (minimum spanning tree) heuristic from iGen to generate the topology. There are 24 links in this topology. All the algorithms perform in the same way (figures not shown), whether or not the network

^{||}<http://www.info.ucl.ac.be/~bqu/igen/>

is properly provisioned. For any source-destination pair, only one path is physically available, so all algorithms have no choice but to choose the same path.

4.2. WELL-CONNECTED TOPOLOGY

Real-world topologies are typically designed to have minimal cost, while being able to stand any link failure. Several methods exist to generate such graphs [3]. We use the Two-Trees heuristic from iGen to generate a graph made of two link-disjoint MSTs. Such a graph remains connected after any single link failure. At least 2 paths exist between any pair of nodes. The topology has 48 links.

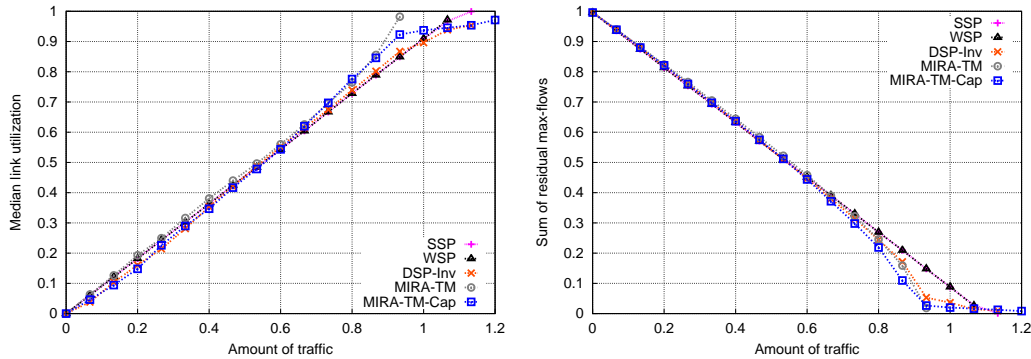


Figure 5: Median link utilization (left) and network-wide residual max-flow (right) in properly provisioned Two-Trees topology.

4.2.1. PROPERLY PROVISIONED NETWORK

Figure 5 shows the median link utilization and the residual max-flow after each connection request on the properly provisioned Two-Trees topology. There is not much difference between the TE algorithms, both in terms of the median link utilization and the sum of the residual max-flows. As the network provisioning matches the traffic matrix, all algorithms manage to reach a high median link utilization (0.93) when the demand equals the network capacity (amount of traffic = 1). It is interesting to note that both SSP and WSP have a linear increase (resp. decrease) of the median link utilization (resp. sum of residual max-flow) until the network is fully loaded. As SSP and WSP use shortest paths that better match to the way the network was provisioned, their behavior is more appropriate in this network than more complex TE algorithms.

4.2.2. IMPROPERLY PROVISIONED NETWORK

Figure 6 shows the median link utilization and the residual max-flow after each connection request on the improperly provisioned Two-Trees topology. In well-provisioned networks, TE algorithms are hardly useful. In networks that have not been provisioned in such a way as to match the traffic demand, the typical low-medium-high load regimes appear quite early, when the median utilization is rather low.

In the low load regime, all algorithms give the same value of the sum of residual max-flow (bottom of Figure 6). In the medium load regime, SSP blocks connection requests while all other algorithms manage to find feasible paths in the network.

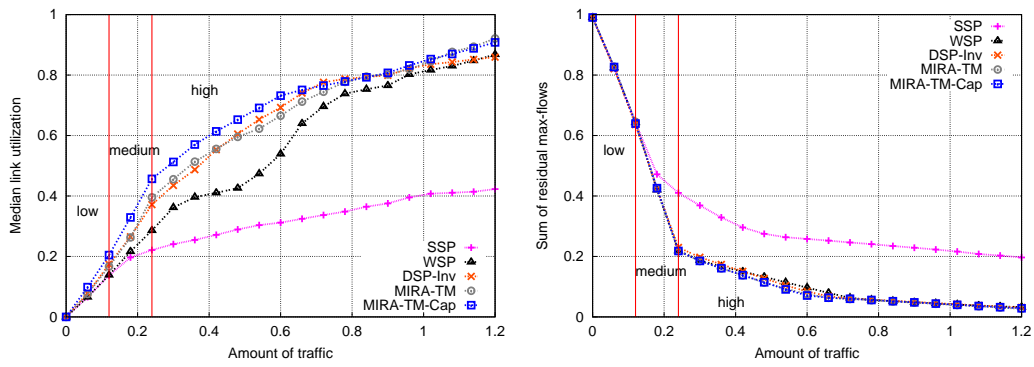


Figure 6: Median link utilization (left) and network-wide residual max-flow (right) in improperly provisioned Two-Trees topology.

4.3. HIGHLY-CONNECTED TOPOLOGY

If local redundancy is required in a topology, any three nodes close to each other can be connected by a triangle. The Delaunay triangulation procedure ensures such a connectivity, leading to locally well-connected graphs. Using the Delaunay heuristic in iGen to connect the 25 nodes, we obtain a topology with 65 links.

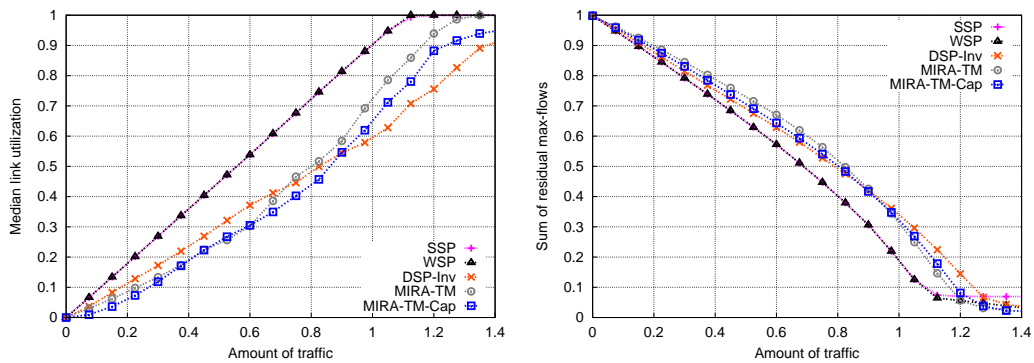


Figure 7: Median link utilization (left) and network-wide residual max-flow (right) in properly provisioned Delaunay topology.

4.3.1. PROPERLY PROVISIONED NETWORK

As this topology provides much redundancy, when assigning link capacities in the properly provisioned scenario, some links that are never used by the shortest paths are assigned a capacity within the same scale as other links (their average). As shown in Figure 7, SSP and WSP have a median link utilization that increases linearly with the amount of traffic until it reaches a value of 1, faster than the other algorithms. DSP-Inv and the MIRA algorithms manage to leave a higher residual max-flow than SSP and WSP. The good provisioning and link redundancy allows all algorithms to use available capacity even when the link utilization becomes high. Contrary to the well-provisioned Two Trees topology, the redundancy of the Delaunay topology allows algorithms to choose very different paths, hence the differences between the algorithms.

4.3.2. IMPROPERLY PROVISIONED NETWORK

Figure 8 shows the median link utilization and network-wide residual max-flow after each connection request on the improperly provisioned Delaunay topology. In the low load regime, DSP-Inv and the two MIRA variants manage to route the requests while leaving a larger residual max-flow than SSP and WSP. SSP keeps a low link utilization at the cost of blocking connection requests, while WSP has the highest median link utilization not to block requests. DSP-Inv and the two MIRA variants manage to both keep a relatively low median link utilization while leaving a residual max-flow larger than WSP.

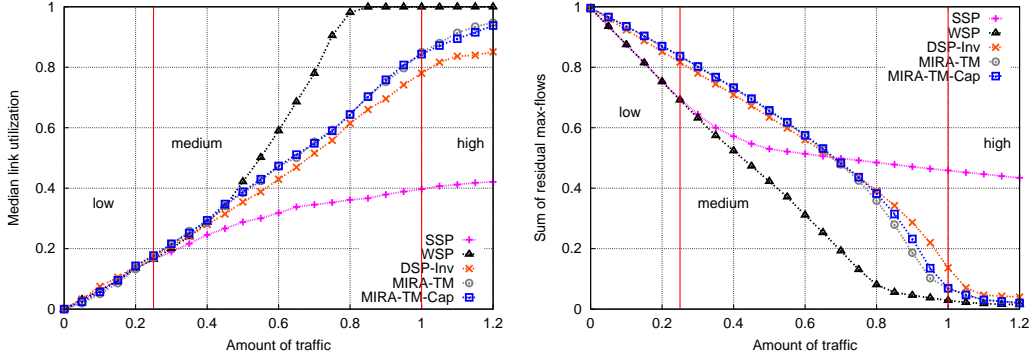


Figure 8: Median link utilization (top) and network-wide residual max-flow (bottom) in the improperly provisioned Delaunay topology.

4.4. DISCUSSION

In Section 3 we studied the behavior of TE algorithms on two real-world networks: Abilene and GEANT. Those two networks have different topological structure, as well as different strategies to set up their IGP weights. The Abilene network is very close to a Minimum Spanning Tree, with 15 undirected links for 12 nodes. The Abilene network has been designed to have minimal number of links, while still being 2-connected, i.e. any single link failure still leaves the network connected. Abilene nodes have node degree 2 or 3. The GEANT network on the other hand is closer to a Two Trees topology than an MST, with 36 undirected links for 22 nodes. GEANT nodes have node degree from 2 to 8, so this network has been designed with far more redundancy than Abilene. Both Abilene and GEANT are over-provisioned networks, the link utilization is pretty low. Over-provisioned is not equivalent to what we call "properly provisioned" in this paper. Proper provisioning means that the capacity of the links in the network match the traffic between all source-destination pairs. In Section 3, we observe that when pushing these networks to their limit, they behave like improperly provisioned networks. It is unlikely that these networks relied on capacity dimensioning to match the traffic demand. Instead, they probably use a simple over-provisioning strategy.

5. CONCLUSION

In this paper, we studied the behavior of on-line TE algorithms under different scenarios. We first relied on two real-world networks, Abilene and GEANT, and their traffic. By scaling their traffic demand, we observed the behavior of TE algorithms when networks are pushed to their limits. We identified three distinct load regimes (low, medium, and high) that correspond to different behaviors of the TE algorithms. In the low load regime, TE algorithms do not provide much benefit compared

to shortest-path routing. In the medium load regime, where shortest-path routing blocks some connection requests, TE algorithms manage to better use the available capacity in the network. Finally, the high load regime corresponds to a situation where not enough capacity is left in the network, so TE algorithms cannot route connection requests. In high load regime, network capacity or redundant links should be added.

In the second part of the paper, we studied the behavior of TE algorithms using synthetic network topologies and traffic demands. We compared the behavior of each type of topology under two network provisioning scenarios. In the first scenario, we provisioned the link capacities as to match the traffic demand. In the second scenario, we provisioned the network without taking into account the traffic demand. As long as the network is properly provisioned, TE algorithms do not provide a significant improvement in using the available capacity, even in highly-connected topologies where many paths are available. When network provisioning does not match the traffic demand, TE algorithms are able to use the redundant links to compensate for the poor provisioning.

We expect that most real-world networks are in a low load regime, where shortest-path routing is good enough. However, as the traffic demand evolves, networks need be updated. One potential further work is to develop a metric that quantifies how badly the current network does not match the current traffic demand. This metric would take into account the network growth and the routing algorithm used, and give insight into when the situation is becoming critical enough so that either TE algorithms should be used, or when the network must be upgraded. We also expect that the topological structure has a non-trivial impact on the behavior of the TE algorithms.

6. ACKNOWLEDGMENTS

Bingjie Fu is supported by the Dutch Technology Foundation STW under the project number DTC.6421. We thank Abilene and GEANT for providing their topology and traffic demands. We are also grateful to Bruno Quoitin, Almerima Jamakovic and Piet Van Mieghem for comments on previous versions of this paper.

REFERENCES

- [1] AWDUCHE, D., CHIU, A., ELWALID, A., WIDJAJA, I., AND XIAO, X. Overview and principles of internet traffic engineering. Internet Engineering Task Force, RFC3272, May 2002.
- [2] BLANCHY, F., MÉLON, L., AND LEDUC, G. An efficient decentralized on-line traffic engineering algorithm for MPLS networks. In *Proc. of ITC-18, Berlin, Germany* (2003).
- [3] CAHN, R. *Wide area network design: concepts and tools for optimization*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [4] CALLON, R. Use of OSI IS-IS for routing in TCP/IP and dual environments. *Request for Comments 1195, IETF* (December 1990).
- [5] CISCO. OSPF design guide. <http://www.cisco.com/warp/public/104/1.html>.
- [6] CLAFFY, K., BRAUN, H., AND POLYZOS, G. Traffic characteristics of the T1 NSFNET backbone. In *INFOCOM* (1993).

- [7] FELDMANN, A., GREENBERG, A., LUND, C., REINGOLD, N., REXFORD, J., AND TRUE, F. Deriving traffic demands for operational IP networks: methodology and experience. In *Proc. of ACM SIGCOMM, Stockholm, Sweden* (2000).
- [8] FOMENKOV, M., KEYS, K., MOORE, D., AND CLAFFY, K. Longitudinal study of internet traffic in 1998-2003. In *Proceedings of WISICT'04* (2004).
- [9] FORTZ, B., REXFORD, J., AND THORUP, M. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine* (October 2002).
- [10] FORTZ, B., AND THORUP, M. Internet traffic engineering by optimizing OSPF weights. In *Proc. of IEEE INFOCOM* (Tel-Aviv, Israel, March 2000).
- [11] GUERIN, R., WILLIAMS, D., AND ORDA, A. QoS routing mechanisms and OSPF extensions. In *Proc. of IEEE Globecom* (1997).
- [12] KODIALAM, M. S., AND LAKSHMAN, T. V. Minimum interference routing with applications to MPLS traffic engineering. In *Proc. of IEEE INFOCOM* (Tel-Aviv, Israel, March 2000).
- [13] KOWALSKI, J., AND WARFIELD, B. Modeling traffic demand between nodes in a telecommunications network. In *Proceedings of ATNAC'95* (1995).
- [14] MANNIE, E. Generalized multi-protocol label switching (GMPLS) architecture. *Request for Comments 3945, IETF* (October 2004).
- [15] MEDINA, A., TAFT, N., SALAMATIAN, K., BHATTACHARYYA, S., AND DIOT, C. Traffic matrix estimation: existing techniques and new directions. In *Proc. of ACM SIGCOMM* (2002), pp. 161–174.
- [16] MOY, J. OSPF version 2. *Request for Comments 2328, IETF* (April 1998).
- [17] NUCCI, A., BHATTACHARYYA, S., TAFT, N., AND DIOT, C. IGP link weight assignment for operational tier-1 backbones. *IEEE/ACM Trans. Netw.* 15, 4 (2007), 789–802.
- [18] ORAN, D. OSI IS-IS intradomain routing protocol. *Request for Comments 1142, IETF* (February 1990).
- [19] PAPAGIANNAKI, K., MOON, S., FRALEIGH, C., THIRAN, P., TOBAGI, F., AND DIOT, C. Analysis of measured single-hop delay from an operational backbone network. In *Proc. of IEEE INFOCOM* (June 2002).
- [20] ROSEN, E., VISWANATHAN, A., AND CALLON, R. Multiprotocol label switching architecture. *Request for Comments 3031, IETF* (January 2001).
- [21] THOMPSON, K., MILLER, G., AND WILDER, R. Wide-Area Internet traffic patterns and characteristics. *IEEE Network magazine* 11, 6 (November/December 1997).
- [22] UHLIG, S. Nonstationarity and high-order scaling in TCP flow arrivals: a methodological analysis. *ACM SIGCOMM Computer Communication Review* 34, 2 (April 2004).
- [23] UHLIG, S., QUOTIN, B., LEPROPRE, J., AND BALON, S. Providing public intradomain traffic matrices to the research community. *ACM SIGCOMM Comput. Commun. Rev.* 36, 1 (2006), 83–86.
- [24] WANG, H., XIE, H., QIU, L., YANG, Y., ZHANG, Y., AND GREENBERG, A. COPE: traffic engineering in dynamic networks. In *Proc. of ACM SIGCOMM'06, Pisa, Italy* (2006).
- [25] ZHANG, Y., ROUGHAN, M., LUND, C., AND DONOHO, D. An information-theoretic approach to traffic matrix estimation. In *Proc. of ACM SIGCOMM* (2003), pp. 301–312.