KTH Electrical Engineering

Towards Robust Traffic Engineering in IP Networks

ANDERS GUNNAR

Licentiate Thesis
Stockholm, Sweden 2007

TRITA-EE 2007:073
ISSN 1653-5146
ISBN 978-91-7178-816-0

KTH School of Electrical Engineering
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framläggs till offentlig granskning för avläggande av teknologie licentiatexamen i telekommunikation måndagen den 10 december 2007 klockan 10.00 i sal Q31, Kungl Tekniska högskolan, Osquldas väg 6, Stockholm.

Tryck: Universitetsservice US AB

Abstract

To deliver a reliable communication service it is essential for the network operator to manage how traffic flows in the network. The paths taken by the traffic is controlled by the routing function. Traditional ways of tuning routing in IP networks are designed to be simple to manage and are not designed to adapt to the traffic situation in the network. This can lead to congestion in parts of the network while other parts of the network is far from fully utilized. In this thesis we explore issues related to optimization of the routing function to balance load in the network.

We investigate methods for efficient derivation of the traffic situation using link count measurements. The advantage of using link counts is that they are easily obtained and yield a very limited amount of data. We evaluate and show that estimation based on link counts give the operator a fast and accurate description of the traffic demands. For the evaluation we have access to a unique data set of complete traffic demands from an operational IP backbone.

Furthermore, we evaluate performance of search heuristics to set weights in link-state routing protocols. For the evaluation we have access to complete traffic data from a Tier-1 IP network. Our findings confirm previous studies who use partial traffic data or synthetic traffic data. we find that optimization using estimated traffic demands has little significance to the performance of the load balancing.

Finally, we device an algorithm that finds a routing setting that is robust to shifts in traffic patterns due to changes in the interdomain routing. A set of worst case scenarios caused by the interdomain routing changes is identified and used to solve a robust routing problem. The evaluation indicates that performance of the robust routing is close to optimal for a wide variety of traffic scenarios.

The main contribution of this thesis is that we demonstrate that it is possible to estimate the traffic matrix with good accuracy and to develop methods that optimize the routing settings to give strong and robust network performance. Only minor changes might be necessary in order to implement our algorithms in existing networks.

To Albin, Arvid and Jenny

Acknowledgments

I would like to thank my advisor Mikael Johansson for his support and inspiration. Without his guidance and encouragement this thesis would never have been written. Many thanks to my co-advisor and former main advisor, Gunnar Karlsson, for believing in me and accepting me as a PhD student. I am also grateful that my manager at SICS, Bengt Ahlgren, provided me with the opportunity to perform research as part of my employment.

I am grateful to Henrik Abrahamsson for being a inspiring colleague but foremost a good friend. My gratitude also goes to Thiemo Voigt for being a great colleague. His comments has greatly improved the quality of this thesis. Many thanks to Adam Dunkels for his help with LaTeX but above all for being a generous and helpful person. Thanks to all members in the NETS lab: Laura Feeney, Björn Grönvall, Ian Marsh and Javier Ubillos for contributing to an inspiring research environment. I am also grateful to Björn Johansson and Pablo Soldati for their help in making this thesis ready for printing.

I have had the opportunity to work with many interesting persons. I would like to express my gratitude to Thomas Telkamp for providing me with traffic data from Global Crossings's global IP backbone network. Thanks to Mattias Söderqvist for his excellent master thesis where he implemented some of the software used in this thesis. It is a bit early to thank Steve Uhlig for a nice discussion at the licentiate seminar. But I would like to express my gratitude to him for his help to allow me to gain access to traffic data from the the GEANT network.

After a hard day at the office it is a relief to come home and play with my two sons, Albin and Arvid. Thanks for being there and forcing me to think about other things than computer networks. Finally, I would like to express my sincere gratitude to my wife Jenny, for her patience and support during my PhD studies. Thanks for being part of my life. I love you.

Contents

Part I

Thesis

Chapter 1

Introduction

Since the launch of the ARPANET in 1969, the network that evolved to what we know as the Internet, has grown at a tremendous rate. Today millions of computers communicate using the Internet and the number of hosts continue to grow. Every day more application and services are deployed on the Internet that has become a pervasive and critical infrastructure.

Originally, the network was designed for researchers sharing research results by using simple services such as email and file transfer. These type of services basically require the network to transport bits from source to destination. However, as the Internet has been adopted by other sectors of society new applications have begun to emerge. Many of these applications such as streamed audio or video and voice transfer, require a higher degree of support from the network and introduce new service requirements such as bounded delay and jitter and limited packet loss. In addition, commercial interests are also incorporated into the provisioning of Internet services. Today the competition between Internet Service Providers (ISPs) makes it more important than ever to reduce the cost of managing the network and optimize the resource use in the network. This poses new requirements on ISPs to manage the traffic situation in an efficient and reliable way to meet service level agreements (SLAs) made with the customers. Hence, new ways to configure the routing and efficient methods to measure and monitor the traffic situation are instrumental for achieving these goals.

The connectionless nature of IP networks make these issues challenging since the transmission rate is regulated from end hosts with limited knowledge of the traffic situation. On the other hand, the path taken by the packets is controlled by the network operator. The operator prefer to keep the routing configuration as static as possible in order to reduce signaling traffic and have the network operate in a predictable manner.

In this thesis we study methods to enhance the routing in the Internet. We investigate methods for capturing the traffic situation using equipment and standards present in routers today, and study the precision of the derivation of the

traffic situation needed in order to make optimization of the routing meaningful. Further, we develop a framework that can be used to understand how the traffic situation is influenced by alterations in the internal routing in the network as well as by events outside the operator's network. The focus is on traffic engineering inside an administrative domain. However, we investigate how changes in the routing between administrative domains affect the traffic situation inside an administrative domain.

1.1 Internet Basics

The Internet is held together by the *Internet Protocol* (IP) that allows data to be interpreted consistently as it travels across the network. Every computer connected to the Internet has a 32-bit IP-address. This address provides a uniform way of identifying the destination in the network. Routers which are the entities that forward traffic from source to destination use the address in the routing decision.

The design philosophy of the Internet is to make the network simple in order to require as little as possible from the underlying networking technology. Instead, most of the complexity needed for communication is placed at the end hosts. The design with a primitive core just forwarding data and complex end hosts is the opposite of the design of telephone networks where all the complexity is placed in the network and the end terminals are kept simple. Another major difference between the Internet and the telephone network is that the Internet is a connectionless communication network while the telephone network is connection-oriented. In a connection-oriented network an end to end path is set up before the sender can start to send data to the receiver. In a connectionless network data is forwarded in packets one hop at the time and each router makes a forwarding decision independently of other routers. Each router has to maintain routing state in order to forward traffic towards the destination. Forwarding is performed by using the information provided in the packet header. This information is used to examine the routing state in the router to look up which outgoing link to forward the packet on.

To simplify the design and isolate implementation changes, the Internet has adopted a layered design. These layers are often described as a stack. Each layer has a specified interface and is responsible for a communication service. How the interfaces are implemented is hidden to other layers. As long as the interface is not altered, implementation changes in the layers are kept isolated inside the layer. The Internet protocol stack is called the TCP/IP reference model after its two most well known protocols. Originally the TCP/IP reference model contained four layers but has evolved to include a fifth layer.

- **Application layer:** This layer contains information about the application at the end host that uses the network to communicate with other applications at other hosts in the network.

- **Transport layer:** The transport layer contains most of the complexity that is needed in order to communicate over a connectionless network. This include congestion control, sequence control, flow control and resending of lost data.
- **Network layer:** The main task of the network layer is routing, i.e. forwarding traffic towards the destination, and maintaining the necessary information to perform this routing.
- **Data link layer:** In the data link layer, traffic is sent over a single hop towards the destination without errors using a noisy channel.
- **Physical layer:** The physical layer is concerned with sending bits over a communication channel. Design issues include coding of bitstreams and delimiters for data packets.

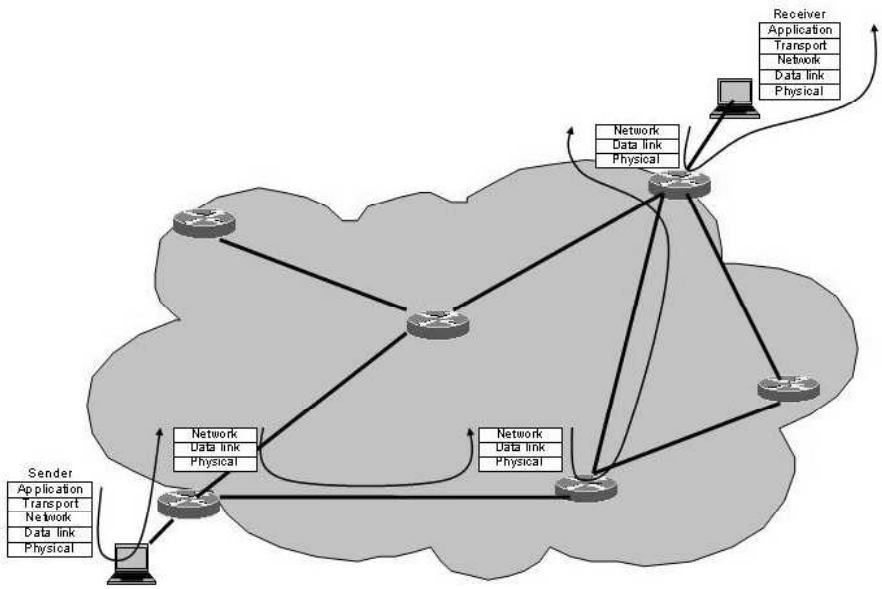

Figure 1.1: An example of how data is transmitted in the Internet

Figure 1.1 shows how data is sent from the source application program to the destination application. A packet leaving the sender node is sent down the protocol stack to the physical layer and is transmitted to the neighboring node. Upon reception at the intermediate node the packet is propagated upward to the network layer where the node detects that it is not the destination of the packet. The destination address in the packet is used as a key in a routing table that keeps track of what outgoing link the packet should be forwarded on. The procedure is repeated hop by hop, until the packet reaches the destination.

1.2 Routing in the Internet

The service provided by an IP network can be described as connectivity. To provision this service a routing mechanism is needed. Since the Internet is managed by different organizations known as Internet Service Providers (ISP) the network is partitioned into subnetworks called Autonomous Systems (AS). Hence, the routing is divided between *intradomain* routing inside an AS, and *interdomain* routing between ASes.

The routing inside an AS is managed by an Interior Gateway Protocol (IGP). Typically, the IGP is a link state routing protocol like Intermediate System Intermediate System (IS-IS) or Open Shortest Path First (OSPF). In link state routing the network is modeled as a graph where nodes represent routers and arcs represent links connecting the routers. Associated with each link is a weight reflecting the cost of sending traffic over the link. Nodes in the network advertise in a Link State Advertisement (LSA) the nodes it has a connection, i.e. neighboring nodes connected with a link. In addition, the LSA includes the link weight. The LSA is flooded in the network to allow each node to collect information about network topology and builds a map of the network. The shortest path to each destination node in the network can be calculated using Dijkstra's algorithm. In this thesis we refer to this type of routing as Shortest Path First (SPF) routing. A variant of shortest path routing is Equal Cost Multi-Path (ECMP) where traffic is split evenly over multiple paths with the same cost to destination.

Although forwarding traffic along shortest paths is simple and easy to implement it has the drawback that it is coarse. The forwarding is based on the destination address only. All traffic from nodes on the path from the source to the destination must follow the same path to the destination. A more fine grained forwarding can be implemented with Multi Protocol Label Switching (MPLS). With MPLS label switch paths (LSP's) are set up between an ingress and egress node pair. The ingress router selects label for an incoming packet based on some criteria such as destination, source/destination or traffic class. Packets following the same paths are grouped in an Forwarding Equivalence Class (FEC). The packet is forwarded along the path based on the label until the packet reaches the egress router of the LSP where the label is removed. Since MPLS allows traffic to be forwarded arbitrarily in the network MPLS has loose restrictions on how paths are calculated. A commonly used approach is to use Constrained Shortest Path First (CSPF). In CSPF links in the network that do not meet a given criteria are removed from the routing calculations. The shortest paths are then calculated in the same manner as in Shortest Path Routing. More sophisticated routing can also be used in conjunction with MPLS. For instance Multi Commodity Flow Optimization (MCNF) [2, 31]. The advantage with MCNF is that the resulting routing setting is optimal for a given objective but is more difficult to implement since traffic is split between more than one MPLS path between ingress and egress routers.

In order to connect the ASes and exchange connectivity information an External Gateway Protocol is used. ISPs usually apply policies reflecting the business

relation between neighboring ASes when exchanging routing information. Typical business relations are customer, provider and peering relations. A customer AS pays a provider AS for connectivity to the rest of the Internet. However, ASes that exchange large amounts of traffic set up peering links to exchange traffic that originates in one AS and is destined to a network in the peering AS or one of its customer ASes. Today there is only a small group of ISPs that are not a customer of another ISP. This group of ISPs, known as *Tier-1* operators, peer with each other in order to get connectivity to the entire Internet.

Policy routing is difficult to implement in link-state routing and reveals details about network topology operators want to keep confidential. Hence, the interdomain routing protocol currently in use is a path vector protocol called Border Gateway Protocol. There an AS announces to its neighboring ASes which networks it has a route to. In order to avoid routing loops a path of ASes is included in the routing messages. If an AS recognizes its own AS number in the path the route is discarded. In addition, the routing announcements have a variety of attributes to express policies associated with announced networks. A detailed description of BGP4 can be found in [17].

1.3 Traffic Engineering

The term *traffic engineering* refers to optimization of network configuration under given network and traffic constraints. This includes transport control to maximize throughput under fairness constraints between users or routing to achieve resilience to router or link failure. However, in the literature traffic engineering is mostly associated with adapting the routing function to the traffic situation to make better use of available network resources.

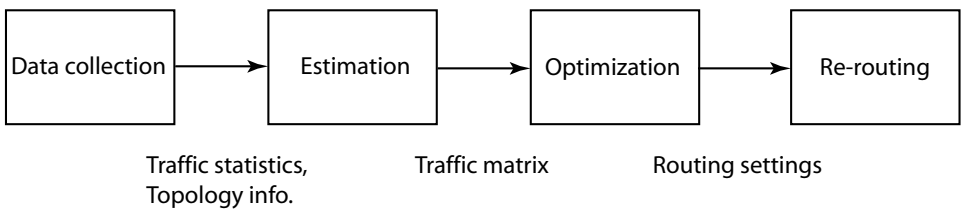

Figure 1.2: The traffic engineering process.

In order to find a suitable routing setting, a number of steps need to be executed. These steps are illustrated in Figure 1.2. The first step is to collect the necessary information about network topology and the current traffic situation. Most traffic engineering methods need as input a traffic matrix describing the demand between each pair of nodes in the network. Obtaining the traffic matrix in a large IP backbone can be a challenging task and the traffic matrix must be estimated from other available data. The traffic matrix together with network constraints

such as network topology and link capacities are used as input to the optimization of the routing. The output from the optimizations need to be translated into parameter values of the routing protocol in use and distributed to the routers.

Omitted in the figure is a feed-back loop from the output to the input of the traffic engineering process. A change in the routing will affect the traffic situation in the network because packets will be routed on different paths due to interactions between inter and intradomain routing. One approach to handle the feed-back loop is to use control theoretic methods to design a routing function that converges to an optimal solution and is stable. We will refer to this approach as *reactive traffic engineering*. Another approach is to find a routing setting that is able to perform well under wide variety of traffic situations. This approach will be referred to as *proactive traffic engineering*. A third alternative is to omit the feed-back loop and regard the traffic situation as independent of the routing; a fair assumption from the perspective of the communication end points. However, an IP backbone is usually not at the end points of a connection and the traffic situation is dependent of the routing. The fact that the dependence is omitted by many researchers is due to the fact that researchers usually do not have access to detailed information about routing configuration and detailed traffic data from operational IP networks. Without proper data it is hard to infer anything conclusive how the traffic situation is affected.

1.4 Characteristics of Internet Traffic

Internet traffic has a rich variety of characteristics depending on location in the network and at what time scale the traffic is observed. For instance, Wide area network and Web traffic have been shown to possess self similar properties (cf. [9, 30]). Basically, self similarity means that traffic behavior is independent of the time scale the traffic is observed. If the traffic is bursty on the milli-second level it is bursty at the second level etc. For instance, Figure 1.3 shows the number of bytes during each 100 millisecond interval of one minute over a link close to the edge of Internet. The plot reveals a clear bursty behavior with periods with large amounts of bytes transmitted interchanged with periods with low traffic intensity. However, it is desirable for a network operator to keep the routing stable in order to avoid oscillatory behavior of the traffic, minimize routing signaling traffic and avoid instability in the routing system. Traffic engineering is preferably performed for a stable traffic situation.

Figure 1.4 shows total traffic in a large backbone during one week. A clear diurnal pattern appears in the plot but there are also fluctuations in the traffic demand. For traffic engineering purposes it is desirable to optimize for a stable peak hour demand but also leave some space for fluctuations in traffic demand.

The total traffic in two subnetworks of a Tier-1 ISP for a 24 hour period is shown in Figure 1.5. At this level of aggregation the random fluctuations in the traffic are small and the traffic is highly predictable.

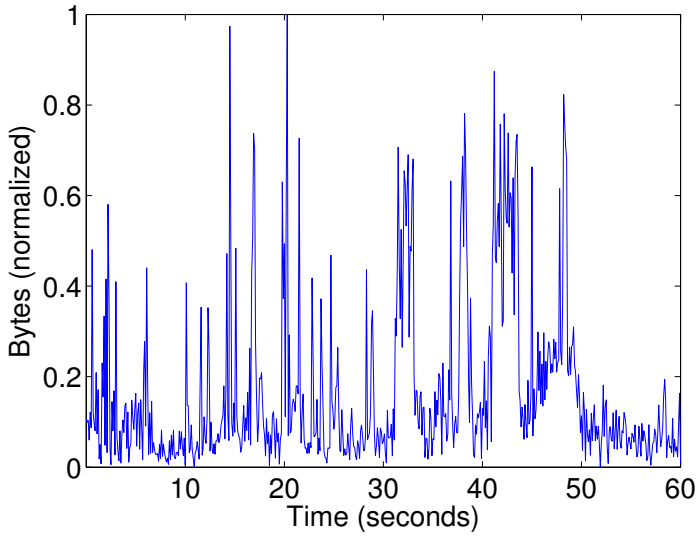

Figure 1.3: Bytes per 100 millisecond sent on a link close to the edge of the Internet during one minute

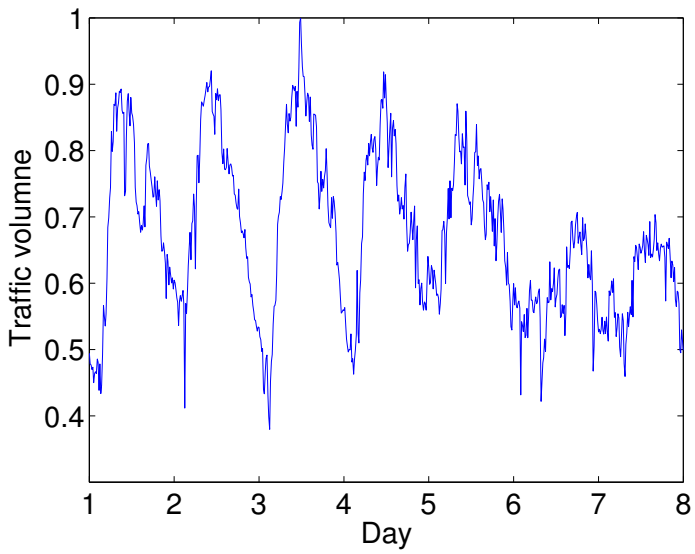

Figure 1.4: Total traffic sent in a large IP backbone for a seven day period (traffic normalized)

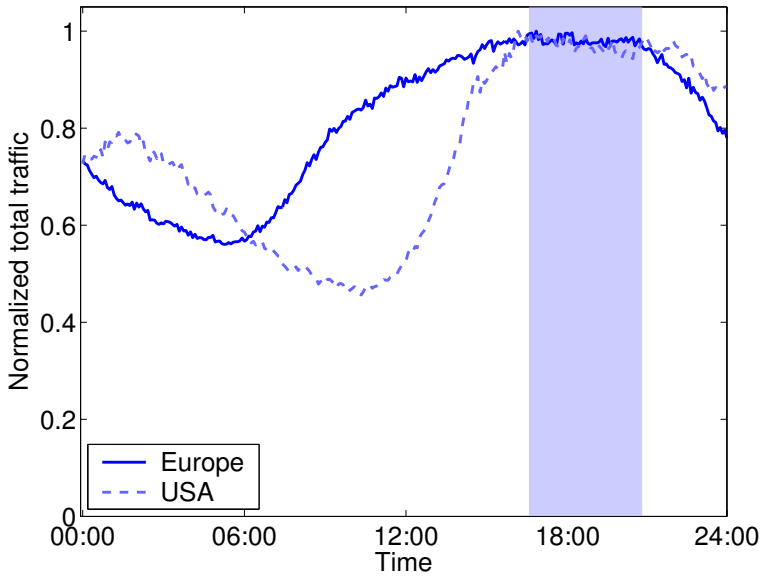

Figure 1.5: Total traffic sent in two subnetworks of a global backbone IP network during a 24 hour period (traffic normalized)

The three figures show traffic observed at different time scales but also at different locations in the network and at different aggregation levels. Figure 1.3 shows traffic behavior close to the edge of the network, Figure 1.4 a large regional IP backbone and Figure 1.5 a global Tier1 IP backbone. We see that traffic is more bursty close to the edge but at higher aggregation levels the traffic is smoother as illustrated in Figure 1.6.

As previously mentioned, network operators strive to keep the routing stable to avoid oscillations and have the traffic situation behave in a more predictable manner. The stability of the traffic demands enables the network operator to make a meaningful prediction about traffic behavior in the future by observing the present traffic situation. Furthermore, the burstiness observed in Figure 1.3 is not only due to the location where the traffic was observed but also due to the time scale of the observation. At time scales of round trip times (10-1000 ms) congestion control is active to alleviate congestion in the network. Traffic engineering on the other hand, is active on longer time scales from seconds to weeks or even years. In Figure 1.5 there is a clear stable behavior of the traffic. The plot in Figure 1.4 reveal a clear pattern for each day in the week but there is also burstiness in the traffic. Nevertheless, we believe it is fair to say that the plots in this section indicate that at the aggregation level and time scales relevant for the problems studied in this thesis there is sufficient stability to optimize the routing function in the network.

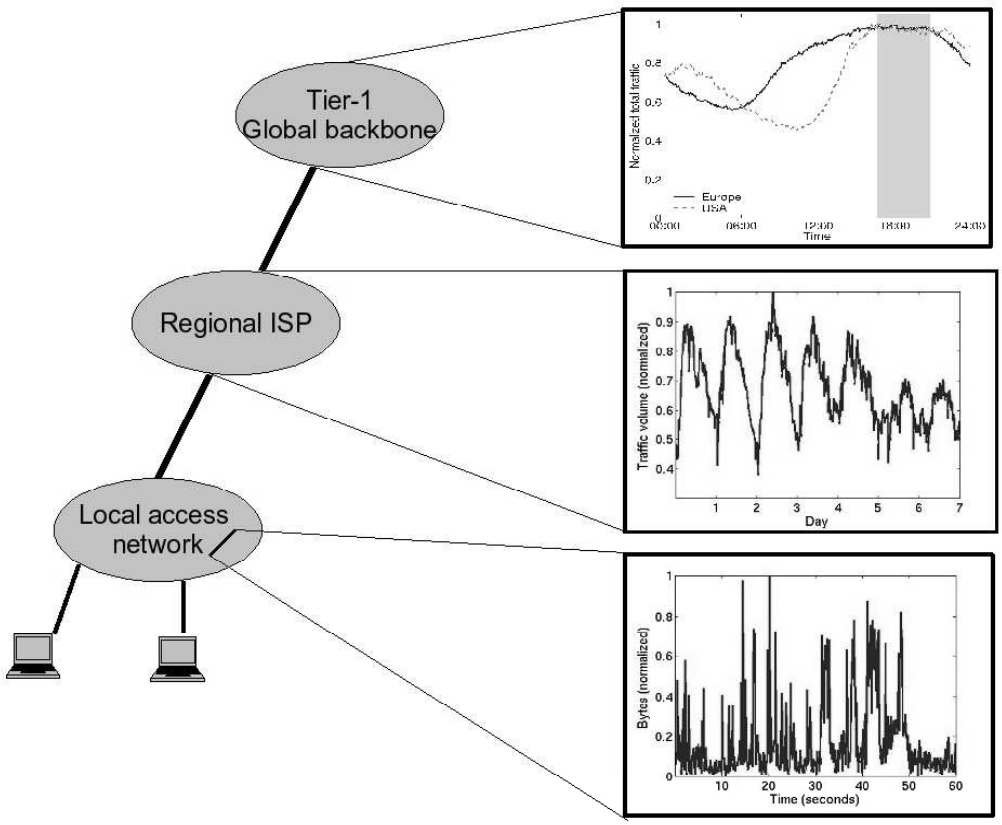

Figure 1.6: Overview of traffic behavior at different aggregation levels

Chapter 2

Scope and Contribution of this Thesis

This section introduces the problems and the approaches used to address the problems in this thesis. The description has an informal character to let the reader gain intuition for the problems and solutions. The scope of related work on the other hand, is wider in order to place the contributions in a context and relate the results to existing knowledge in the field. Detailed results and discussions follow in the included papers.

2.1 Notation

We consider a network with N nodes and L directed links. Such a network has $N(N-1)$ pair of distinct nodes that may communicate with each other. The aggregate communication rate between any pair (n, m) of nodes is called the *point-to-point demand* between the nodes and denoted by s_{nm}. The matrix $S = [s_{nm}]$ is called the *traffic matrix*. For our purposes it is more convenient to represent the traffic matrix in vector form by enumerating all source-destination pairs, letting s_k denote the point-to-point demand of node pair k, and introducing $s = [s_k]$ to be the vector of demands for all source-destination pairs.

While ordinary SPF routing forwards traffic along a single path between source and destination, more advanced forwarding mechanisms such as MPLS allow for multiple paths between source and destination. To this end, we let Π_k be the set of paths between source-destination pair k, and let $\alpha_{\pi k}$ represent the fraction of s_k sent over path π. We assume that all traffic is assigned to some path, i.e.

$$\sum_{\pi \in \Pi_k} \alpha_{\pi k} = 1 \tag{2.1}$$

The paths can be summarized in a *routing matrix* $R \in \mathbb{R}^{L \times P}$ with entries

$$r_{lk} = \sum_{\pi \in \Pi_k} \rho_{l\pi} \alpha_{\pi k} \tag{2.2}$$

where $\rho_{l\pi}$ is an indicator variable taking value one if link l is part of path π, and zero otherwise. Note that for SPF routing, $\alpha_{\pi k} \in \{0, 1\}$ so each column of the routing matrix has ones on the entries corresponding to the links in the single path between source and destination, and zeros on all other entries. In general, however, $\alpha_{\pi k}$ and hence r_{lk} are real numbers.

2.2 Estimation of the Traffic Matrix

Conducting large scale flow measurements in an IP backbone to obtain the traffic matrix can be challenging task. An alternative approach is to estimate the traffic matrix from link load measurements that are readily obtained from Simple Network Measurement Protocol (SNMP). To find the desired traffic demands we need to establish a link between the measured link loads and the unknown point-to-point traffic demands. This link is the routing configuration encoded in the routing matrix R. The traffic demands s and link loads t are related via

$$Rs = t \tag{2.3}$$

The *traffic matrix estimation problem* is simply the one of estimating the non-negative vector s based on knowledge of R and t. The challenge in this problem comes from the fact that this system of equations tends to be highly underdetermined: there are typically many more source-destination pairs ($\mathcal{O}(N^2)$) than links in a network ($\mathcal{O}(N)$), and (2.3) has many more unknowns than equations. Is is only in rare instances that the routing matrix will have full rank. One such example is when the network is fully meshed and traffic is routed on the single-hop paths connecting the communicating node pair. In general, however, networks are far from fully meshed and since the number of links tend to grow linearly while the the number of node pairs grow quadratically, the traffic estimation problem will become even more underconstrained as the size of the network grows.

Figure 2.1 illustrates the challenge in the traffic matrix estimation problem using a simple example. The Figure shows a simple network with three nodes and three traffic demands. From the picture it is clear that looking at the link loads alone, it is impossible to observe an increase in s_{13} if s_{12} and s_{23} decrease at the same time. For clarity we explicitly state (2.3) for the example in Figure 2.1:

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} s_{12} \\ s_{13} \\ s_{23} \end{pmatrix} = \begin{pmatrix} t_{12} \\ t_{23} \end{pmatrix}$$

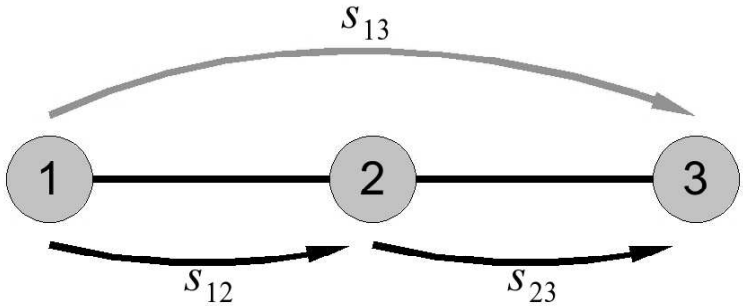

Figure 2.1: A simple network with three nodes and three traffic demands

The rows in the routing matrix above represent l_{12} and l_{23} , and columns represent the paths π_{12}, π_{13} and π_{23}. It is clear that the routing matrix does not have full rank, and that the null space is spanned by the direction $(1, -1, 1)$.

To solve the estimation problem more information about the traffic must be added. This can be a prior guess $s^{(p)}$ of the traffic situation, or a model of the traffic (e.g. that the traffic matrix is a sample from a given probability distribution). One could then try to find the traffic matrix closest to the prior guess that explains the observed link loads, see Figure 2.2 This can be formulated as the optimization problem:

$$\begin{array}{ll} \text{minimize} & d(\hat{s}, s^{(p)}) \\ \text{subject to} & R\hat{s} = t \\ & \hat{s} \succeq 0 \end{array} \tag{2.4}$$

where $\hat{s}$ denotes an estimate of s and $d(\hat{s}, s^{(p)})$ the distance (in an appropriate measure) between $\hat{s}$ and $s^{(p)}$.

In many cased, however, it makes sense to sacrifice some accuracy in explaining the link loads in order to have a better match with the prior guess. One then solves the problem:

$$\begin{array}{ll} \text{minimize} & d(\hat{s}, s^{(p)}) + \lambda \|R\hat{s} - t\| \\ \text{subject to} & \hat{s} \succeq 0 \end{array} \tag{2.5}$$

This formulation is sometimes referred to as *regularization* (cf. [5]). The nonnegative weight λ is called the regularization parameter, and allows to emphasize good reconstruction of the observed link loads or good accordance with the prior guess.

For this formulation, the traffic matrix estimation problem now breaks down to picking the prior guess, the appropriate distance measure $d(\cdot, \cdot)$, and the regularization parameter λ. To illustrate the regularized approach, we plot the spatial

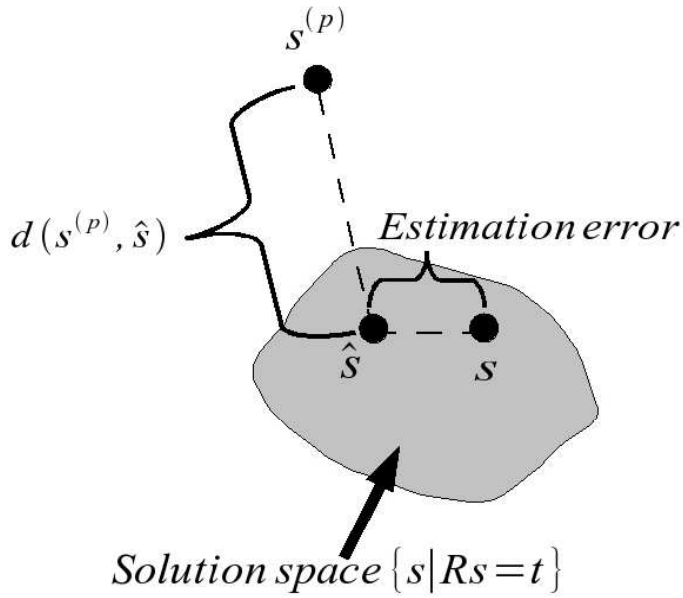

Figure 2.2: The relation between prior guess and estimated traffic demands and real traffic demands

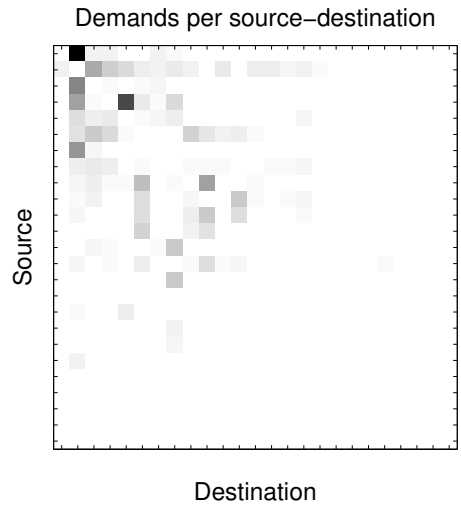

Figure 2.3: Spatial distribution of real traffic demands from a large IP backbone. Source nodes sorted in descending order.

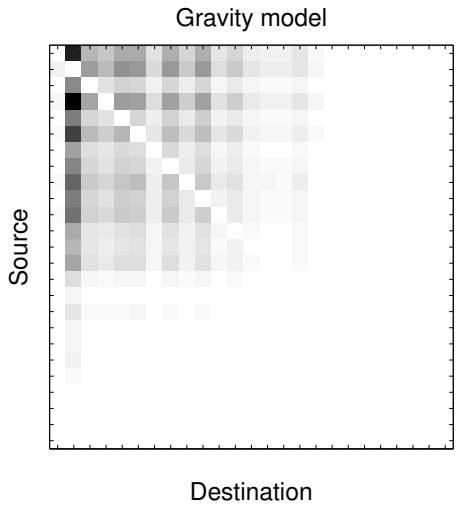

Figure 2.4: Spatial distribution of traffic demands for gravity prior. Source nodes sorted in descending order for real traffic demands.

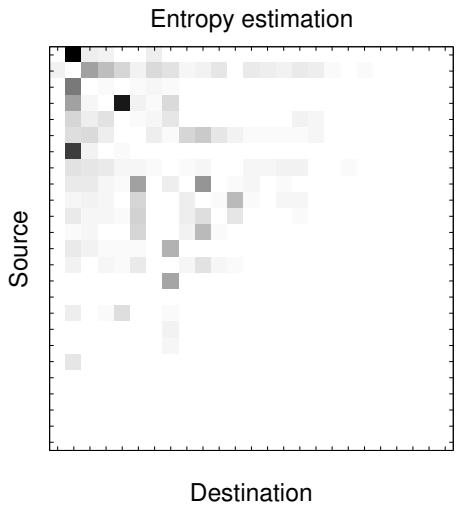

Figure 2.5: Spatial distribution of estimated traffic demands using entropy estimation. Source nodes sorted in descending order for real traffic demands.

distribution of actual measured traffic demands in Figure 2.3. A prior guess about the traffic behavior (this particular prior is based on the gravity model [48]) shown in Figure 2.4 is clearly not very accurate. However, by the appropriate choice of distance measure (here the Kullback-Leibler divergence) and the regularization parameter results in an estimate shown in Figure 2.5 which is rather close to the true traffic matrix.

In paper A in this thesis we evaluate a wide selection of regularized methods and discuss new approaches to the problem. We evaluate our results using a unique data set of complete traffic matrices from an operational Tier-1 IP backbone.

2.3 Search Heuristics for OSPF/IS-IS Routing

The weights setting in an OSPF or IS-IS network will influence the routing performance as they determine along which paths the traffic is routed. The *OSPF/IS-IS weight setting problem* consists of assigning positive integer weights to links in order to achieve better network performance when the demands are routed according to the rules of the OSPF or the IS-IS protocols.

However, the restrictions of the OSPF/IS-IS protocols makes the problem of finding weights that optimizes the routing NP-hard [14]. To make the optimization run faster a search heuristic can be applied to the problem. The heuristic attempts to improve an objective function by evaluating different OSPF weights. As input to the heuristic we need a graph $G = (N, L)$ and a traffic matrix S. The output consists of a set of weights, where each weight is associated with an arc in the graph. The heuristic generate a sequence of new weights using a local search. Each set of link weights is viewed as a point in a high-dimensional search space. A neighbor to a point is another set of weights produced by changing the value of one (or sometimes more) weights. Different local searches generate different neighbors and evaluate these with respect to the overall performance objective. The neighbor with the best objective is the one that is used in the next iteration of the algorithm. The algorithm is typically terminated either when no improvement is detected or after a specified number of iterations.

The network shown in Figure 2.6 has four nodes and four bidirectional links with a capacity of 10 units of traffic in each direction. There are two traffic demands, s_{14} transmits 7.5 units of traffic between nodes 1 and 4, where s_{34} transmits 2.5 units of traffic between nodes 3 and 4. In case a) s_{14} is routed on path 1-3-4 and s_{34} is routed on the path 3-4 leading to 100% utilization of link l_{34}. Many search heuristics attempts to alleviate congestion by deviating traffic from the link with highest utilization. In our example the weight of link l_{34} is increased in b) deviating s_{14} to the path 1-2-4. Congestion is lowered to 75% on link l_{12} and l_{24}. Finally, some search heuristics attempt to balance load on path equal cost paths (ECMP). In Figure 2.6 c) demand s_{14} is split between the paths 1-2-4 and 1-3-4 leading to maximum link utilization of 63% on link l_{34}. The example in Figure 2.6 highlights

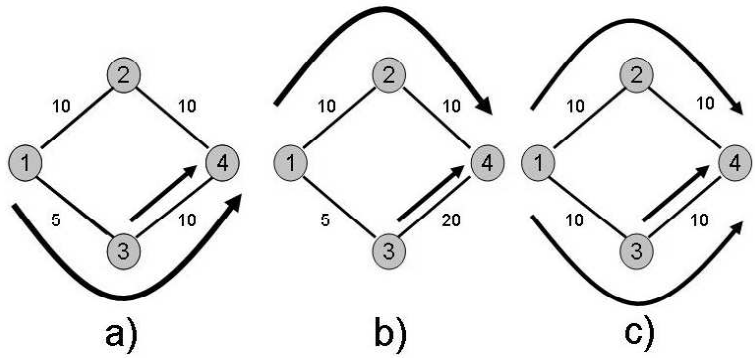

Figure 2.6: A simple example of search heuristics for finding a weight setting

the limitations of SPF and ECMP forwarding. The best solution would be to split demand s_{14} with 2/3 of the traffic on path 1-2-4 and 1/3 of traffic on path 1-3-4 reducing highest link utilization to 50%. However, this split ratio is not possible with SPF or ECMP forwarding.

In this thesis we have implemented two search heuristics. The first is called Strictly Descending Search and was first suggested by Ramakrishnan and Rodrigues [32]. The heuristic is a greedy algorithm that in each step tries to find the weight change that gives the largest improvement. In each iteration a link is selected and the link weight is increased such that one traffic demand is deviated from the link. This is performed for every link in the network and the link change that gives the largest improvement executed. The algorithm terminates when there is no improvement on any link in the network. The second heuristic is the well known search heuristic by Fortz and Thorup [14]. This algorithm is a variant of a class of heuristics known as *tabu search* (cf. [31]). Tabu search maintains a list of the search history which is forbidden (tabu) in the following iterations of algorithm. To avoid that the same weight setting is evaluated several times, a hash table maps a weight setting to a slot in the table. Initially all slots are set to zero. When a weight setting is evaluated its corresponding hash slot is set to one. If a weight setting maps to a slot set to one the weight setting is not evaluated further. In addition, only a random set of neighbors are evaluated. The set of evaluated neighbor is divided by three every time the objective is improved and multiplied by two every time it is not improved.

In paper B in this thesis we evaluate two search heuristics for weight setting in link state routing using data from a Tier-1 IP backbone. The paper also studies how the heuristics perform using estimated traffic demands.

2.4 Robust Routing in MPLS Enabled Networks

A system that is able to cope with variations from the normal operating conditions is said to be *robust*. In a networking context this entails the ability to sustain acceptable performance despite foreseeable traffic variations and component failures. A common optimization objective in robust networking is to minimize the worst-case link loads, where worst-case should be understood as over all potential load variations or component failures.

Within an Autonomous System (AS) load shifts occur due to several reasons, such as router of link failure or shifting user behavior. Another reason which to a large extent is beyond the network operators control is load shifts due to interdomain reroutes. A multihomed AS may receive routes to the same destination network from more than one location. In this case the interdomain routing protocol, BGP, selects one route according to a specified decision process.

The first step is to determine if there is a route to the egress point of the AS. Next BGP examines a number of BGP specific attributes. If BGP still is unable to select one route, the shortest distance according to intradomain routing is considered. This is sometimes referred to as hot-potato routing [41]. The final step is to use a vendor-specific tie-breaking. Figure 2.7 illustrates a simple example of a situation where a prefix is announced by two routers. In the example router R3 selects the route announced by R2 since it has the shortest IGP distance to R2. However, if the route announced by R2 is withdrawn the traffic towards network 192.168.0.0/16 injected in the network by R3 is shifted from the route announced by R2 to the route announced by R1, causing a potentially massive change of load on the links in the network.

The limitations of SPF forwarding based on destination address makes it difficult to implement a robust routing setting in an IP network. Instead, MPLS forwarding offers an opportunity to implement a routing that is able to cope with a wide variety of traffic scenarios.

Several methods for robust routing have been proposed recently [3, 4, 18, 37]. In this thesis we base our developments on the approach by Ben-Ameur and Kerivin [4] as we find it the most transparent. The robust routing problem can be formulated as the following optimization problem:

$$\begin{array}{ll} \text{minimize} & u_{\max} \\ \text{subject to} & \displaystyle\sum_{k}\sum_{\pi\in\Pi_k} \rho_{l\pi}\alpha_{\pi k}s_k \le c_l u_{\max}\,\forall l, \forall s \in \mathcal{S} \\ & \displaystyle\sum_{\pi\in\Pi_k} \alpha_{\pi k} = 1, \qquad \alpha_{\pi k} \ge 0 \end{array} \tag{2.6}$$

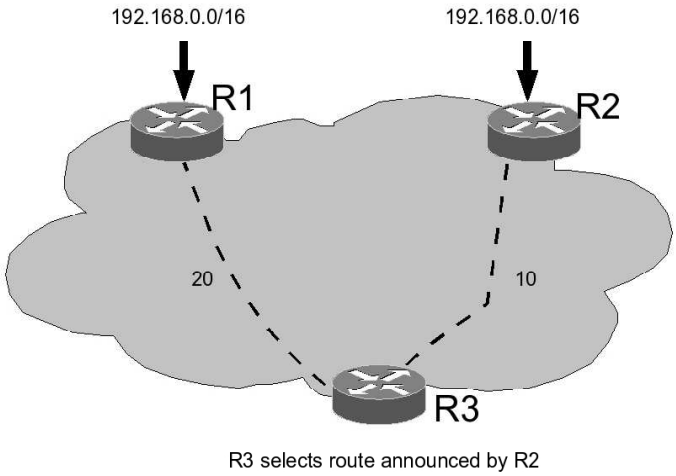

Figure 2.7: Routing scenario where the prefix 192.168.0.0/16 is announce by two peering points in the network. Router R3 has to select a route using the BGP decision process.

The first set of constraints state that the total traffic across each link l is bounded by the link capacity times the maximal link utilization for each traffic scenario in $\mathcal{S}$, while the second constraint states that all traffic must be routed across some path.

The classical way of solving (2.6) is by column generation. Rather than explicitly enumerating all paths in the network, one starts out with a small subset of paths (*e.g.*, the shortest-hop routing) and then sequentially adds new paths to the problem to improve the optimization objective, see *e.g.*, [31] for details. To reduce the computational burden of accounting for all potential traffic scenarios, we proceed in a similar fashion as column generation is used to avoid explicit enumeration of all paths: one starts out with a single traffic scenario in the traffic scenario set $\mathcal{S}$, solves the routing problem, and then verifies whether the computed routing satisfies the link constraints for all feasible traffic loads. If this is not the case, one adds the traffic matrix that violates the constraints the most to the vertex description of the uncertainty set and repeats. The resulting method is a combined column- and constraint generation scheme, and is readily shown to have finite convergence (*e.g.* [4]).

In Paper C in this thesis we address the problem of robust routing under changes in the interdomain routing. The routing setting found by the optimization can be implemented by MPLS and is optimized for all admissible traffic changes due to interdomain routing changes.

2.5 Related Work

Since the first communication networks were built operators have been tuning the routing function in order to accommodate more traffic. However, it was not until the Internet boom during the late 1990's that it became an important research area for IP networks. This chapter surveys research in the area during recent years. We start with methods for deriving the traffic situation in the network and continue with methods to optimize the routing.

Methods for Obtaining the Traffic Matrix

The methods for deriving the traffic matrix can be divided into three classes. The first class is estimation based where the traffic demands are estimated from incomplete data. The second class of methods are measurement based and rely on flow measurements performed in routers. The third class is a combination of measurements and estimation.

Estimation Based Methods

The origin-destination estimation problem for telephone traffic is a well-studied problem in the telecom world. For instance, already in 1937, Kruithof [20] suggested an iterative method for estimation of point-to-point traffic demands in a telephone network based on a prior traffic matrix and measurements of incoming and outgoing traffic. Kruithof's method was first analyzed by Krupp [21], who showed that the approach can be interpreted from an information theoretic point-of-view: it minimizes the Kullback-Leibler distance from the prior guess of the traffic matrix. Further, Krupp showed that the extended iterative method convergences to the unique optimal solution. It is interesting to note that Kruithof's method appears to be the first iterative scaling method in statistics, and that these methods are closely related to the EM-algorithm [10].

However, it appears that it was not until 1996 that the problem was addressed specifically for IP networks. To handle the difficulties of an under-constrained problem, Vardi [42] assumed a Poisson model for the traffic demands. Using the Posisson model the sample average and sample covariance of the link loads are calculated for a sequence of measurements. The samples are used as additional constraints. The traffic demands are estimated by Maximum Likelihood estimation. Related to Vardi's approach is Cao *et al.* [6] who propose to use a more general scaling law between means and variances of demands. The Poisson model is also used by Tebaldi and West [38], but rather than using ML estimation, they use a Bayesian approach. Since posterior distributions are hard to calculate, the authors use a Markov Chain Monte Carlo simulation to simulate the posterior distribution. The Bayesian approach is refined by Vaton *et al.* [43], who propose an iterative method to improve the prior distribution of the traffic matrix elements. The estimated traffic matrix from one measurement of link loads is used in the next

estimation using new measurements of link loads. The process is repeated until no significant change is made in the estimated traffic matrix. An evaluation of the methods in [38, 42] together with a linear programming model is performed by Medina *et al.* [23]. A novel approach based on choice models is also suggested in the article. The choice model tries to estimate the probability of an origin node to send a packet to a destination node in the network. Similar to the choice model is the gravity model introduced by Zhang *et al.* [48]. In its simplest form the gravity model assumes a proportionality relation between the traffic entering the network at node i and destined to node j and the total amount of traffic entering at node i and the total amount of traffic leaving the network at node j. The authors of the paper use additional information about the structure and configuration of the network such as peering agreements and customer agreements to improve performance of the method. An information-theoretic approach is used by Zhang *et al.* [49] to estimate the traffic demands. Here, the Kullback-Leibler divergence is used to minimize the mutual information between source and destination. In all papers mentioned above, the routing is considered to be constant. In a paper by Nucci *et al.*[25] routing is changed and shifting of link load is used to infer the traffic demands.

Measurement Based Methods

An alternative method to estimation for finding the traffic demands in a network is to use the measurement facilities present in routers, e.g. Cisco's Netflow. Feldmann *et al.* [13] collect flow measurements from routers using Cisco's Netflow tool and derive point-to-multipoint traffic demands using routing information from inter and intradomain routing protocols. Choi and Bhattacharyya [8] investigate the accuracy of sampled Netflow. The authors of the paper find that accuracy is satisfactory but care should be taken when Netflow is used in backbone routers since measurement overhead grows linearly with the number of active flows passing the router. An approach to control the measurement overhead is developed by Duffield *et al.* [24]. A scaling factor is recalculated in order to control sampling rate and number of flow records dynamically. In addition, the method is designed to minimize variance in the estimator. Estan *et al.* [12] discuss improvements to Netflow but the changes are aimed to facilitate traffic flow analysis and are not directed towards traffic matrix measurements.

Combined Traffic Matrix Derivation

A more recent approach is to combine measurement based traffic matrix derivation with estimation-based methods. Papagiannaki *et al.* [29] use Netflow measurements over a 24 hour period to calibrate parameters of a fanout model. The fanout model assumes that the fraction of traffic destined to each other node in the network stays stable even though the corresponding traffic demands fluctuates over time. Each router in the network performs the necessary measurements to

calibrate the fanout factors that are sent to the Network Operations Center (NOC). Link-count measurements performed by SNMP are used by the NOC to derive the traffic matrix. The authors devise a heuristic to check the parameters of the fanout model in order to monitor the accuracy of the measurements. If the measured values differ significantly from the parameter values the model is re-calibrated. Traffic matrix estimation methods are divided into three generations by Soule *et al.* [35]. The first generation is constituted by methods where additional constraints are added to the estimation by calculating sample covariance over a time series of link-load measurements [6, 42]. The second generation consists of regularized methods [48, 49]. The third generation uses flow measurements together with estimation to obtain the traffic matrix. Soule *et al.* introduce three methods from the third generation. The first is the fanout method described above. In addition, two novel methods are introduced. The PCA method is based on principal component analysis and attempts to find a low dimension representation of the traffic demands. Lakhina *et al.* observe in [22] that a traffic matrix is dominated by a limited number of flows. By concentrating the analysis of the traffic matrix to these eigenflows the problem is reduced to a well-posed estimation problem.

Traffic Engineering

The aim of traffic engineering is to optimize the usage of network resources under traffic constraints. However, the traffic situation in the network may change over time, e.g. due to changing user behavior, new applications or changes in the routing system. To handle the changes there are basically two approaches.

Proactive traffic engineering aims to configure the routing such that it is able to cope with a large variety of traffic situations. The operation of the network is simple and controllable but performance will not be optimal in some situations.

Reactive traffic engineering solutions, on the other hand, continuously monitors the state of the network and adapts the routing to handle changes in the traffic situation. This approach enables the network to handle unanticipated changes and the network to operate at an optimal (or at least favorable) point at all times. However, this requires the network operator to monitor the state of the network which imposes extra overhead.

Proactive Traffic Engineering

In link state routing the link weight is the parameter the operator can adjust to balance load in the network. One of the earliest and most referenced papers on link weight optimization is due to Fortz and Thorup [14]. The authors use a search heuristic which is shown to be very efficient in finding a suitable weight setting to a given traffic situation. The search heuristic is extended to find a weight setting for a wider range of traffic situations in [15]. Ramakrishnan and Rodrigues [32] use a different heuristic that increases a link weight until one of the paths traversing the link finds a shorter path to the destination and is deviated. If the

change leads to lower link utilization the change is executed and another link is selected. Traffic engineering using search heuristics with estimated traffic matrices is explored by Roughan *et al.* [33]. Wang *et al.* [46] compute the link weights from the solution of the dual problem of a multi commodity flow problem. Variables in the dual problem can be interpreted as cost of utilizing the resource associated with the dual variable; in our case a link in the network. A somewhat different approach is taken by Sridharan *et al.* [36]. Instead of calculating the link weights the authors use a heuristic to allocate routing prefixes to equal-cost multi-paths.

Xu *et al.* [47] introduce DEFT where traffic can be sent over non shortest paths using exponential penalty on longer paths. DEFT can be integrated in OSPF/IS-IS routing with minor changes only.

Applegate and Cohen [3] show that it is possible to find an efficient routing setting with fairly limited knowledge of the traffic demands. Furthermore, the authors give a lower bound on performance for the routing for all possible traffic situations. Column generation is used by Ben-Ameur and Kerivin [4] to find a routing that is optimal for a set of different traffic matrices. The authors describe an algorithm which starts from a small set of paths in the network and set of traffic scenarios and continue to add paths and traffic scenarios until no further improvement is observed for the objective. It is shown that the algorithm terminates in a finite number of steps to an optimal solution. In a recent article Wang *et al.* [45] propose Common-case Optimization with Penalty Envelope (COPE) which computes a routing setting that optimizes for a set of traffic matrices which constitute common case traffic scenarios. Furthermore, COPE gives an upper bound of performance of a larger set of admissible traffic scenarios called a traffic envelope.

Abrahamsson *et al.* [1] use a two step cost function which strives to keep load in the network below a given utilization set by the network operator. The method combines properties of cost functions that minimize link utilization with cost functions that minimize bandwidth usage in the network.

Reactive Traffic Engineering

One of the earliest papers is Gallager's classical paper on minimum delay routing [16] where the author gives sufficient conditions for minimum delay routing and develops a distributed algorithm to calculate the minimum delay routing. The distributed algorithm is dependent on a global traffic dependent parameter for convergence which makes the algorithm impractical for implementation. Vutukury and Garcia-Luna-Aceves [44] devise an algorithm that approximates the results of Gallagers distributed algorithm.

Reactive traffic engineering with MPLS has been the subject of a number of research papers during recent years. Elwalid *et al.* [11] introduce a routing algorithm based on optimization. A distributed method called TeXCP for MPLS traffic engineering is introduced by Kandula *et al.* [19]. Load balancing is performed over a set of precomputed MPLS paths between source and destination based on feed-

back about the traffic situation from the network. The authors prove stability and convergence as well as optimality of the method.

2.6 Contributions in this Thesis

In this thesis we have investigated and benchmarked methods to obtain the traffic matrix by estimation from link load measurements. Contrary to previous studies, that have used a partial traffic matrix or demands estimated from aggregated Netflow traces [23, 49], we use a unique data set of complete traffic matrices from a global IP network measured over five-minute intervals. This allows us to do an accurate data analysis on the time-scale of typical link-load measurements and enables us to make a balanced evaluation of different traffic matrix estimation techniques. We explore some novel approaches to the problem and show that methods which rely on second order moments have poor performance due to slow convergence of the estimation of the covariances. The analysis indicate that regularized optimization from link load measurements give an accurate estimate of the traffic situation. The advantage of regularized methods is that they are simple to implement, efficient to execute, do not require resource consuming measurements which produce large volumes of traffic data that need to be sent over the network for processing.

We investigate two weight setting methods on a network topology and traffic data from a commercial IP network. The connection between estimation and traffic engineering has to the best of our knowledge not been studied on a network with complete traffic data. Fortz and Thorups search heuristic has only been studied on partial traffic data. Descending search first introduced by Ramakrshnan and Rodriguez has as far as we have found not been studied for a real IP network with real data.

The intra and interdomain routing systems were originally designed to be independent of each other. However, in practice routing decisions made in intradomin routing influence interdomain routing decisions. In a series of papers Teixeira *et al.* [40, 41] study and model these interactions. The interaction between intra and interdomain routing is omitted in most research on intradomain routing even though it can give rise to massive changes in the traffic matrix [39]. In this thesis we demonstrate that it is possible to find an intradomain routing setting that allows performance to be close to optimal under all admissible traffic changes due to interdomain routing changes. This routing setting can be realized with legacy protocols with minor changes in hardware or software.

Chapter 3

Summary of Papers Included in this Thesis

This thesis is composed of three redistributed papers, paper A, paper B and paper C. All three papers have been published in international conferences or workshops with peer review. Paper A was awarded best student paper at the conference.

Paper A: Traffic Matrix Estimation on a Global IP Backbone - A Comparison on Real Data

A. Gunnar, M. Johansson and T. Telkamp. Traffic Matrix Estimation on a Global IP Backbone - A Comparison on Real Data. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, October 2004, Taormina Italy.

Summary: In this paper we consider the problem of estimating the point-to-point traffic matrix in an operational IP backbone. The analysis is based on complete traffic matrices from a global IP network measured over five-minute intervals. The paper describes the data collection infrastructure, present spatial and temporal demand distributions, investigate the stability of fan-out factors, and analyze the mean-variance relationships between demands. We evaluate existing and novel methods for traffic matrix estimation, including recursive fanout estimation, worst-case bounds, regularized estimation techniques, and methods that rely on mean-variance relationships. We discuss weaknesses and strengths of the various methods. We highlight differences in traffic patterns on different continents and show how this affect the estimation.

This paper was awarded the best student paper award at the conference.

A preliminary version of this paper can be found in "Traffic Matrix Estimation for a Global IP Network" at the Nordic teletraffic seminar, Oslo Norway August 2004.

Contribution of this paper: The contribution of this work is a balanced evaluation of traffic matrix estimation methods using a unique data set of complete

traffic matrices from an operational IP backbone.

My contribution: I implemented the methods in close cooperation with Mikael Johansson and performed a large part of the analysis of the data set and the experimental evaluation.

Paper B: Performance of Traffic Engineering in Operational IP-Networks - An Experimental Study

A. Gunnar, H. Abrahamsson and M. Söderqvist. Performance of Traffic Engineering in Operational IP-Networks - An Experimental Study, In T. Magedanz, E.R.M. Madeira and P. Dini Editors: *IPOM 2005, LNCS 3751*, pp 202-211 Springer Verlag.

Summary: Today, the main alternative for intra-domain traffic engineering in IP networks is to use different methods for setting the weights (and so decide upon the shortest-paths) in the routing protocols OSPF and IS-IS. In this paper we study how traffic engineering perform in real networks. This paper analyzes different weight-setting methods and compare performance with the optimal solution given by a multi-commodity flow optimization problem. Further, the robustness in terms of how well they manage to cope with estimated traffic matrix data is investigated. The evaluation is performed using network topology and traffic data from an operational IP network.

Parts of this work can be found in "Performance of Traffic Engineering using Estimated Traffic Matrices". In proceedings of Radio Sciences and Communication RVK 05, June 2005, Linköping Sweden.

Contribution of this paper: The contribution of this work is an evaluation of two search heuristics for weight setting in OSPF/IS-IS using complete traffic data from a Tier-1 IP network operator.

My contribution: I performed the analysis in the paper and performed most of the writing of the paper. The implementation was performed by Mattias Söderqvist but I made some adjustments to the code in order to fit the experiments in the paper.

Paper C: Robust Routing Under BGP Reroutes

A. Gunnar and Mikael Johansson. Robust Routing Under BGP Reroutes, In *Proceedings of Globecom 2007*, November 2007, Washington DC, USA.

Summary: Configuration of the routing is critical for the quality and reliability of the communication in a large IP backbone. Large traffic shifts can occur due to changes in the interdomain routing that are hard to control by the network operator. In this paper we describe a framework for modeling potential traffic shifts due to BGP reroutes to calculate worst-case traffic scenarios. The worst case traffic scenarios are used to find a single routing configuration that is robust against all possible traffic shifts due to BGP reroutes. The benefit of our approach is illustrated using BGP routing updates and network topology from an operational IP

network. Our experiments demonstrate that the robust routing is able to obtain a consistently strong performance under large interdomain routing changes.

A similar approach was used in "Data-driven traffic engineering: techniques, experiences and challenges". In proceedings of Broadnets 2006, San Jose, California, USA.

Contribution of this paper: The main contribution of this paper is the design and evaluation of an algorithm to calculate a routing setting that is robust to shifts in traffic patterns caused by interdomain routing changes.

My contribution: I formulated the problem of combining information about interdomain routing and traffic demands with intradomain routing optimization. The solution approach with combined column and constraint generation emerged from discussions with my advisor. I refined, implemented and evaluated the algorithms, and wrote the main part of the paper.

Other Publications by the Author not Included in this Thesis

This section contains a list of peer reviewed publications authored or co-authored by the author of this thesis. The author changed family name from Andersson to Gunnar in August 2003.

- A. Gunnar, B. Ahlgren, O. Blume, L. Burness, P. Eardley, E. Hepworth, J. Sachs and A. Surtees, Access and Path Selection in Ambient Networks. In *Proc. IST Mobile Summit 2007*, 1-5 July 2007, Budapest, Hungary.

- A. Gunnar, Identifying Critical Traffic Demands in an IP Backbone. In *Proc. Swedish National Computer Networking Workshop, SNCNW 2006*, 26-27 Oct 2006, Luleå, Sweden.

- M. Johansson and A. Gunnar, Data-driven traffic engineering: techniques, experiences and challenges. In *Proc. Broadnets 2006*, 1-5 October 2006, San Jose, California.

- M. Brunner, A. Galis, L. Cheng, J. Colas, B. Ahlgren, A. Gunnar, H. Abrahamsson, R. Szabo, S. Csaba, J. Nielsen, S. Schuetz, A. Gonzalez, R. Stadler and G. Molnar, Towards Ambient Networks Management. In *Proc. IEEE MATA 2005 Second International Workshop on Mobility Aware Technologies and Applications*, November 2005, Montreal, Canada.

- M. Söderqvist and A. Gunnar, Performance of Traffic Engineering using Estimated Traffic Matrices. In *Proc. Radio Sciences and Communication RVK'05*, June 2005, Linköping, Sweden.

- H. Abrahamsson and A. Gunnar, Traffic Engineering in Ambient Networks : Challenges and Approaches. In *Proc. Swedish National Computer Networking Workshop, SNCNW 2004*, November 2004, Karlstad, Sweden.

- M. Brunner, A. Galis, J. Colas, Jorge, A. Gunnar, B. Ahlgren, H. Abrahamsson, R. Szabo, S. Csaba, J. Nielsen, A. Gonzalez, R. Stadler, G. Molnar and L. Cheng, Ambient Networks Management Challenges and Approaches. In *Proc. IEEE MATA 2004 1st International Workshop on Mobility Aware Technologies and Applications*, November 2004, Florianopolis, Brazil.

- A. Gunnar, M. Johansson and T. Telkamp, Traffic Matrix Estimation for a Global IP Network. In *Proc. 17th Nordic Teletraffic Seminar*, August 2004, Oslo, Norway.

- H. Abrahamsson, B. Ahlgren, J. Alonso, A. Andersson and P. Kreuger, A Multi Path Routing Algorithm for IP Networks Based on Flow Optimisation. In *Proc. QofIS'02*, October 2002, Zürich, Switzerland.

- B. Ahlgren, A. Andersson, O. Hagsand, and I. Marsh, Dimensioning links for IP telephony, In *Proc. 2nd IP-Telephony Workshop (IPtel 2001)*, April 2001, New York City, New York, USA.

Chapter 4

Discussion and Future Directions

Traffic engineering attracted a lot of attention form researchers and industry at the turn of the millennium. At the time it was expected that new services would emerge and cause congestion in the Internet. New ways to tune the routing would be needed in order to accommodate the expected growth in traffic volumes. However, the new bandwidth demanding services never emerged and the recession that followed after the turn of the millennium made much of the fiber unused. Hence, acquiring new capacity has not been costly for ISPs. Nevertheless the appearance of and new applications such as peer to peer file sharing or YouTube and TV distributed of over IP (IPtv) has started a rapid increase of bandwidth demand. To tune the routing will be important for network operators to save costs but also to make the network more robust to sudden changes in traffic patterns.

In this thesis we have shown that it is possible to monitor the traffic situation and optimize the routing in IP backbone networks using legacy protocols. Furthermore, by taking intradomain routing decisions into account we are able to find a routing setting that is able to have network utilization perform close to optimal for all admissible traffic patterns due to intradomain routing changes which often are beyond the operator's control. The focus in this thesis has been on large IP backbones. Even though IP backbones span over large geographical areas, sometimes world wide, they usually contain a limited number of routers and links. Hence, it is possible for a human being to grasp and gain intuition about how the network should be monitored and operated. Furthermore, the cost of upgrading the network must be considered minor in the light of the communication capacity of optical fiber links [27]. Internet traffic at the backbone level is highly predictable and planning the management and upgrading of the network is possible. On the other hand network traffic can behave in an unpredictable manner in case of for instance router or link failure [39]. This calls for methods for monitoring the traffic situation and optimizing the routing function in order to deliver a reliable communication service to the customers. Furthermore, even if average utilization in the network is low [26] it is a well known fact that traffic in the Internet is far from

uniformly distributed (cf. [13, 28]) leading to a large fraction of the network being underutilized while a small number of links with high utilization. Balancing load on these critical links can lead to significant performance gains in the network and delay upgrading of the network.

In the early days of the Internet routing could be adapted to the traffic situation [34]. However, this was soon abandoned due to oscillatory behavior of the routing. Nowadays the routing configuration is set to a static value and is rarely changed. Reactive traffic engineering on the other hand, requires new functionality to be installed in routers in the network. Our aim in this thesis has been to optimize legacy functionality in the network as much as possible. However, new software of hardware may be needed in order to split flows arbitrarily between several paths between source and destination. This can be achieved by adopting the solutions described in [1, 7].

The structure of networks in the past has been a meshed core where edge routers connect to the core in a tree structure with only one path for the traffic to take to reach the rest of the Internet. This has lead to a research focus in traffic engineering on backbone networks since this is the region where there is more than one possible route to the destination. However, in future networks we expect more path diversity closer to the edge of the network. More path diversity at the edge rises the possibility of traffic engineering in this region of the network as well as in the core. The bursty traffic behavior closer to the edge together with a wider diversity of link technologies poses new challenges and possibly new solutions to traffic engineering in this region.

The topic of this thesis has been on aspects of intradomain traffic engineering. The problem of intradomain traffic engineering is a much more complex problem. For instance in intradomain routing it is assumed that all participating entities are cooperation and charing a common goal. In intradomain routing on the other hand, the participating entities are competitors as well as cooperating to achieve a common goal. How this traffic engineering is affected by this is not very well understood and needs further research.

Bibliography

[1] H. Abrahamsson, J. Alonso, B. Ahlgren, A. Andersson, and P. Kreuger. A multi path routing algorithm for IP networks based on flow optimisation. In *Proc. Third COST 263 International Workshop on Quality of Future Internet Services, QoFIS 2002*, pages 135–144, Zürich, Switzerland, October 2002. Springer. LNCS 2511.

[2] R. K. Ahuja, T. L. Magnati, and J.B. Orlin. *Network Flows*. Prentice Hall, 1993.

[3] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *Proc. ACM SIGCOMM*, pages 313–324, Karlsruhe, Germany, August 2003.

[4] W. Ben-Ameur and H. Kerivin. Routing of uncertain demands. *Optimization and Engineering*, 6(3):283–313, 2005.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] J. Cao, D. Davis, S. Vander Wiel, and B. Yu. Time-varying network tomography: router link data. *Journal of Americal Statistical Association*, 95:1063–1075, 2000.

[7] Z. Cao, Z. Wang, and E. Zegura. Performance of hashing-based schemes for internet load balancing. In *Proc. IEEE INFOCOM*, pages 332–341, Tel-Aviv, Israel, 2000.

[8] B. Choi and S. Bhattacharrya. On the accuracy and overhead of cisco sampled netflow. In *Proc. ACM Sigmetrics Workshop on Large-Scale Network Inference (LSNI)*, Banff, Canada, June 2005.

[9] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE /ACM Transactions on Networking*, 5(6):835–846, 1997.

[10] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions, Suppl. 1*, Supplement Issue No. 1:205–237, 1984.

[11] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS adaptive traffic engineering. In *Proc. IEEE INFOCOM*, pages 1300–1309, Anchorage, Alaska, USA, May 2001.

[12] C. Estan, K. Keys, D. Moore, and G. Varghese. Building a better netflow. In *Proc. ACM SIGCOMM*, pages 245–256, Portland, Oregon, USA, August 2004.

[13] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: Methodology and experience. In *Proc. ACM SIGCOMM*, pages 257–270, Stockholm, Sweden, August 2000.

[14] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proc. IEEE INFOCOM*, pages 519–528, Tel-Aviv, Israel, March 2000.

[15] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, 2002.

[16] R. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, COM-25(1):73–85, 1977.

[17] S. Halabi and D. McPherson. *Internet Routing Archtectures*. Cisco Press, 2001.

[18] M. Johansson and A. Gunnar. Data-driven traffic engineering: techniques, experiences and challenges. In *Proc. Broadnets 2006*, San Jose, Califonia, USA, October 2006.

[19] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: responsive yet stable traffic engineering. In *Proc. ACM SIGCOMM*, pages 253–264, Philadelphia, Pennsylvania, USA, 2005.

[20] J. Kruithof. Telefoonverkeersrekening. *De Ingenieur*, 52(8):E15–E25, 1937.

[21] R. S. Krupp. Properties of Kruithof's projection method. *The Bell System Technical Journal*, 58(2):517–538, February 1979.

[22] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proc. ACM SIGMETRICS*, pages 61–72, New York, USA, June 2004.

[23] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *Proc. ACM SIGCOMM*, pages 161–174, Pittsburgh, Pennsylvania, USA, August 2002.

[24] M. Thorup N. Duffield, C. Lund. Learn more, sample less: control of volume and vairance in network measurement. *IEEE Transactions in Information Theory*, 51(5):1756–1775, 2005.

[25] A. Nucci, R. Cruz, N. Taft, and C. Diot. Design of IGP link weight changes for estimation of traffic matrices. In *Proc. IEEE INFOCOM*, pages 2341–2351, Hong Kong, March 2004.

[26] A. Odlyzko. Networks are lightly utilized, and will stay that way. *Review of Network Economics*, 2(3):210–237, September 2003.

[27] A. Odlyzko. The evolution of price discrimination in transportation and its implications for the internet. *Review of Network Economics*, 3(3):323–346, September 2004.

[28] K. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, and C. Diot. A pragmatic definition of elephants in internet backbone traffic. In *Proc. ACM Internet Measurment Workshop*, pages 175–176, Marseille, France, November 2002.

[29] K. Papagiannaki, N. Taft, and A. Lakhina. A distributed approach to measure IP traffic matrices. In *Proc. ACM Internet Measurement Conference*, pages 161–174, Taormina, Italy, October 2004.

[30] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[31] M. Pioro and D. Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, 2004.

[32] K.G. Ramakrishnan and M.A. Rodrigues. Optimal routing in shortest-path data networks. *Bell Labs Technical Journal*, 6(1):117–138, 2001.

[33] M. Roughan, Mikkel Thorup, and Yin Zhang. Traffic engineering with estimated traffic matrices. In *Proc. ACM Internet Measurement Conference*, pages 248–258, Miami Beach, Florida, USA, October 2003.

[34] A. Shaikh, J. Rexford, and K. G. Shin. Load-sensitive routing of long-lived ip flows. In *Proc. ACM SIGCOMM*, pages 215–226, Cambridge, Massachusetts, USA, 1999.

[35] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot. Traffic matrices: Balancing measurements, inference and modeling. In *Proc. ACM SIGMETRICS*, pages 362–373, June 2005.

[36] A. Sridarhan, R. Guerin, and C. Diot. Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks. In *Proc. IEEE INFOCOM*, San Francisco, Califonia, USA , November 2003.

[37] A. Sridharan, R. Guérin, C. Diot, and S. Bhattacharyya. The impact of traffic granuarity of robustness of traffic aware routing. Technical report, University of Pennsylvania, 2004.

[38] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):557–576, June 1998.

[39] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan. Traffic matrix reloaded: The impact of routing changes. In *Proc. Passive Active Measurements*, Boston, Massachusetts, USA, 2005.

[40] R. Teixeira, T. Griffin, G. Voelker, and A. Shaikh. Network sensitivity to hot potato disruptions. In *Proc. ACM SIGCOMM*, pages 231–244, Portland, Oregon , USA, August 2004.

[41] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. In *Proc. ACM SIGMETRICS*, pages 307–319, New York, USA, June 2004.

[42] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the Americal Statistical Association*, 91(433):365–377, March 1996.

[43] S. Vaton and A. Gravey. Network tomography : an iterative bayesian analysis. In *Proc. ITC 18*, Berlin, Germany, August 2003.

[44] S. Vutukury and J. J. Garcia-Luna-Aceves. A simple approximation to minimum-delay routing. In *Proc. ACM SIGCOMM*, pages 227–238, Cambridge, Massachusetts, USA, 1999.

[45] H. Wang, H. Xie, L. Qiu, Y. Yang, Y. Zhang, and A. Greenberg. Cope: traffic engineering in dynamic networks. In *Pric. ACM SIGCOMM*, pages 99–110, Pisa, Italy, 2006.

[46] Y. Wang, Z. Wang, and L. Zhang. Internet traffic engineering without full mesh overlaying. In *Proc. IEEE INFOCOM*, pages 565–571, Anchorage, Alaska, USA, May 2001.

[47] D. Xu, M. Chiang, and J. Rexford. Deft: Distributed exponentially-weighted flow splitting. In *Proc. IEEE INFOCOM*, pages 71–79, Anchorage, Alaska, USA, May 2007.

[48] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proc. ACM SIGMETRICS*, pages 206–217, San Diego, Califonia, USA, June 2003.

[49] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information theoretic approach to traffic matrix estimation. In *Proc. ACM SIGCOMM*, pages 301–312, Karlsruhe, Germany, August 2003.

Part II

Included Papers

Chapter 5

Paper A: Traffic Matrix Estimation on a Large IP Backbone - a Comparison on Real Data

Anders Gunnar, Mikael Johansson, and Thomas Telkamp. In *Proceedings of the Third ACM SIGCOMM Conference on Internet Measurements IMC 2004*, Taormina, Sicily, Italy, October 2004.

Abstract

This paper considers the problem of estimating the point-to-point traffic matrix in an operational IP backbone. Contrary to previous studies, that have used a partial traffic matrix or demands estimated from aggregated Netflow traces, we use a unique data set of complete traffic matrices from a global IP network measured over five-minute intervals. This allows us to do an accurate data analysis on the time-scale of typical link-load measurements and enables us to make a balanced evaluation of different traffic matrix estimation techniques. We describe the data collection infrastructure, present spatial and temporal demand distributions, investigate the stability of fan-out factors, and analyze the mean-variance relationships between demands. We perform a critical evaluation of existing and novel methods for traffic matrix estimation, including recursive fanout estimation, worst-case bounds, regularized estimation techniques, and methods that rely on mean-variance relationships. We discuss the weaknesses and strengths of the various methods, and highlight differences in the results for the European and American subnetworks.

5.1 Introduction

Many of the decisions that IP network operators make depend on how the traffic flows in their network. A *traffic matrix* describes the amount of data traffic transmitted between every pair of ingress and egress points in a network. When used together with routing information, the traffic matrix gives the network operator valuable information about the current network state and is instrumental in traffic engineering, network management and provisioning (see, e.g., [1, 18, 2, 17]).

Despite the importance of knowing the traffic matrix, the support in routers for measuring traffic matrices is poor and operators are often forced to estimate the traffic matrix from other available data, typically link load measurements and routing configurations. In its simplest form, the estimation problem then reduces to finding a non-negative vector s that satisfies $Rs = t$, where R is a matrix reflecting the routing, t is a vector of measured link loads and s is a vectorized version of the (unknown) traffic matrix. The link loads are readily obtained using the Simple Network Management Protocol (SNMP). This approach leads to an under-constrained problem since the number of links in a network is typically much smaller than the number of node pairs. Some sort of side information or assumptions must then be added to make the estimation problem well-posed.

To evaluate how well different approaches to traffic matrix estimation will work in an operational IP network, and how reasonable various assumptions are, one needs access to a measured traffic matrix on the time-scale of standard link-load measurements. Previous studies have used NetFlow data to measure the traffic matrix in 5-minute increments on a single router [3] or one-hour traffic matrices on a partial network [23]. However, since NetFlow data is unable to capture traffic variability within flows, this is not very accurate for validating estimation methods that use a time-series of link-load measurements. Our study provides new results in the sense that it uses a complete network traffic matrix, based on direct measurements at 5-minute intervals. The data set is collected from Global Crossing's global backbone and consists of routing configuration and the number of bytes transfered in MPLS-tunnels during 5-minute intervals over a 24-hour period.

To make the analysis more transparent, we extract traffic matrices and routing information for the American and European subnetworks. We present temporal and spatial demand distributions and analyze some statistical properties of the demands. In particular, we find that there is a surprisingly strong relationship between the mean and variance of demands, and that fanout factors tend to be relatively more stable over time compared to the demand themselves. We then evaluate a selection of existing methods for traffic matrix estimation, including gravity models, regularized methods (such as Bayesian and maximum entropy approaches), and methods that exploit mean-variance relationships. In addition, we investigate the use of worst-case bounds and estimation of fanout factors based on a time-series of link load measurements. We find that the regularized methods work very well provided that we choose the regularization parameter,

i.e., the tradeoff between prior information and link measurement, appropriately. Somewhat surprisingly, we fail to achieve good results using methods that exploit mean-variance relationship. We argue that the failure stems from the problem of accurately estimating the covariance matrix of link loads, and present a study on synthetic data to support our claim.

One can note that many classes of traffic matrices occur in the literature (see [11] for a thorough classification). In this paper, we only study the performance of the estimation methods on PoP-to-PoP traffic matrices. This choice is solely based on properties of the data we have obtained, and we make no statement on which class of traffic matrices is more important than the other.

The remaining parts of this paper are organized as follows. In Section 5.2, we present related work in this area. Section 5.3 introduces the problem and notation. The estimation methods that we evaluate are introduced in Section 5.4, while data collection, data analysis and benchmarking of the methods is presented in Section 7.4. Finally, some concluding remarks are collected in Section 6.5.

5.2 Related work

The origin-destination estimation problem for telephone traffic is a well-studied problem in the telecom world. For instance, already in 1937, Kruithof [9] suggested a method for estimation of point-to-point traffic demands in a telephone network based on a prior traffic matrix and measurements of incoming and outgoing traffic. However, it appears that it was not until 1996 that the problem was addressed specifically for IP networks. In order to handle the difficulties of an under-constrained problem, Vardi [20] assumes a Poisson model for the traffic demands and covariances of the link loads is used as additional constraints. The traffic demands are estimated by Maximum Likelihood estimation. Related to Vardi's approach is Cao *et al.* [3] that propose to use a more general scaling law between means and variances of demands. The Poisson model is also used by Tebaldi and West [19], but rather than using ML estimation, they use a Bayesian approach. Since posterior distributions are hard to calculate, the authors use a Markov Chain Monte Carlo simulation to simulate the posterior distribution. The Bayesian approach is refined by Vaton *et al.* [21], who propose an iterative method to improve the prior distribution of the traffic matrix elements. The estimated traffic matrix from one measurement of link loads is used in the next estimation using new measurements of link loads. The process is repeated until no significant change is made in the estimated traffic matrix. An evaluation of the methods in [19, 20] together with a linear programming model is performed by Medina *et al.* [13]. A novel approach based on choice models is also suggested in the article. The choice model tries to estimate the probability of an origin node to send a packet to a destination node in the network. Similar to the choice model is the gravity model introduced by Zhang *et al.* [23]. In its simplest form the gravity model assumes a proportionality relation between the traffic entering the network at node i and

destined to node j and the total amount of traffic entering at node i and the total amount of traffic leaving the network at node j. The authors of the paper use additional information about the structure and configuration of the network such as peering agreements and customer agreements to improve performance of the method. An information-theoretic approach is used by Zhang *et al.* in [24] to estimate the traffic demands. Here, the Kullback-Leibler distance is used to minimize the mutual information between source and destination. In all papers mentioned above, the routing is considered to be constant. In a paper by Nucci *et al.* [15] the routing is changed and shifting of link load is used to infer the traffic demands. Feldmann *et al.* [7] uses a somewhat different approach to calculate the traffic demands. Instead of estimating from link counts they collect flow measurements from routers using Cisco's NetFlow tool and derive point-to-multipoint traffic demands using routing information from inter- and intra-domain routing protocols.

5.3 Preliminaries

Notation and Problem Statement

We consider a network with N nodes and L directed links. Such a network has $P = N(N-1)$ pair of distinct nodes that may communicate with each other. The aggregate communication rate (in bits/second) between any pair (n, m) of nodes is called the *point-to-point demand* between the nodes, and we will use s_{nm} to denote the rate of the aggregate data traffic that enters the network at node n and exits the network at node m. The matrix $S = [s_{nm}]$ is called the *traffic matrix*. It is usually more convenient to represent the traffic matrix in vector form. We then enumerate all P source-destination pairs, and let s_p denote the point-to-point demand of node pair p.

For simplicity, we will assume that each point-to-point demand is routed on a single path. The paths are represented by a *routing matrix* $R \in \mathbb{R}^{L \times P}$ whose entries r_{lp} are defined as

$$r_{lp} = \begin{cases} 1 & \text{if the demand of node pair } p \text{ is routed across link } l \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Note that the routing matrix may easily be transformed to reflect a situation where traffic demands are routed on more than one path from source to destination by allowing fractional values in the routing matrix. Let t_l denote the aggregate data rate on link l, $t = [t_l] \in \mathbb{R}^L$ be the vector of link rates, and $s \in \mathbb{R}^P$ be the vector of demands for all source-destination pairs. Then, s and t are related via

$$Rs = t \tag{5.2}$$

The *traffic matrix estimation problem* is simply the one of estimating the non-negative vector s based on knowledge of R and t.

The challenge in this problem comes from the fact that this system of equations tends to be highly underdetermined: there are typically many more source-destination pairs than links in a network, and (5.2) has many more unknowns than equations.

It is important to note that in an IP network setting, not all links are interior links connecting the core routers in the network: some of the links are access and peering links that supply data to and receive data from the edge nodes. To make this more explicit, we introduce the notation $e(n)$ for the link over which demand enters at node n, and $x(m)$ for the link over which demand exits at node m. For ease of notation, we assume that each edge node is either an access or a peering point (if this is not the case we can always introduce artificial nodes in our network representation so that this holds). Under these assumptions, $t_{e(n)}$ is the total traffic entering the network at node n and $t_{x(m)}$ is the total traffic exiting the network at node m. Finally, we let $\mathcal{A}$ be the set of nodes acting as access points, and $\mathcal{P}$ the set of nodes acting as peering points.

Alternative formulations of traffic estimation problems

The traffic matrix as a demand distribution

Since demands are non-negative, it is natural to normalize s with the total network traffic

$$s_{\text{tot}} = \sum_i \sum_j s_{ij} = \sum_n t_{e(n)}$$

and view $\tilde{s} = s/s_{\text{tot}}$ as a probability distribution. We may then interpret $\tilde{s}_p$ as the probability that a random packet in the network is sent between node pair p. Introducing $\tilde{t} = t/s_{\text{tot}}$, we can re-write (5.2) as

$$\begin{cases} R\tilde{s} = \tilde{t} \\ \mathbf{1}^T \tilde{s} = 1, \quad \tilde{s} \succeq 0 \end{cases} \tag{5.3}$$

The traffic estimation problem then becomes the one of estimating a vector $\tilde{s}$ that satisfies (5.3) based on knowledge of R and $\tilde{t}$ (cf. [9, 10]).

Fanout formulations

Another alternative is to normalize the demands by the total aggregate traffic entering the source node, i.e., to write

$$s_{nm} = \alpha_{nm} \sum_m s_{nm} = \alpha_{nm} t_{e(n)} \qquad \sum_m \alpha_{nm} = 1 \tag{5.4}$$

Rather than estimating s_{nm}, one can now focus on estimating the *fanouts* $\alpha_{nm} = s_{nm}/t_{e(n)}$. Also the fanouts can be interpreted as probability distributions: α_{nm} is

the probability that a random packet entering the network at node n will exit the network at node m (cf. [13, 12]).

5.4 Methods for Traffic Matrix Estimation

Gravity Models

A simple method for estimating the traffic matrix is to use a so-called *gravity model*. Although these models have a long history in the social sciences [25] and in telephony networks [8], the first application to demand estimation in IP networks appears to be [16]. In our notation, the basic version of the gravity model predicts the demand between node n and node m as

$$s_{nm}^{(p)} = Ct_{e(n)}t_{x(m)} \tag{5.5}$$

where C is a normalization constant that makes the sum of estimated demands equal to the measured total network traffic. With the choice $C = 1/\sum_m t_{x(m)}$, the gravity model reduces to

$$s_{nm}^{(p)} = \frac{t_{x(m)}}{\sum_m t_{x(m)}} t_{e(n)}$$

and a comparison with (5.4) reveals that this is equivalent to the fanout model

$$\alpha_{nm} = \frac{t_{x(m)}}{\sum_m t_{x(m)}}$$

i.e., that the amount of data that node n sends to node m is proportional to the fraction of the total network traffic that exits at node m. Such a model makes sense if the user populations served by different nodes are relatively uniform. However, as pointed out in [23], traffic transit between peering networks behaves very differently. This has led to the generalized gravity model, where traffic between peers is forced to be zero, i.e.,

$$s_{nm}^{(p)} = \begin{cases} 0 & \text{if } n \in \mathcal{P} \text{ and } m \in \mathcal{P} \\ Ct_{e(n)}t_{x(m)} & \text{otherwise} \end{cases}$$

Once again, C is a normalization constant that makes, for example, the estimated total traffic equal to the measured total network traffic. In this study, however, we focus on the simple gravity model and leave the generalized gravity model without further reference. It should be noted that the gravity model does not use any information about the traffic on links interior to the network, and that the estimates are typically not consistent with the link load measurements (in fact, the model may not even produce consistent estimates of the total traffic exiting each node). Thus, gravity models are often not used in isolation, but in combination with some statistical approach that accounts for measured link loads. Such methods will be described next.

Statistical Approaches

Kruithof's Projection Method

One of the oldest methods for estimating traffic matrices is the iterative method due to Kruithof [9]. The original formulation considers the problem of estimating point-to-point traffic in a telephony network based on a known prior traffic matrix and measurements of total incoming and total outgoing traffic to each node in the network. Thus, Kruithof's method can, for example, be used to adjust the gravity model estimate to be consistent with measurement of total incoming and outgoing traffic at edge nodes.

Kruithof's method was first analyzed by Krupp [10], who showed that that the approach can be interpreted from an information theoretic point-of-view: it minimizes the Kullback-Leibler distance from the prior traffic matrix $\left[s_{ij}^{(p)}\right]$ (interpreted as a demand distribution). Krupp also extended Kruithof's basic method to general linear constraints,

$$\begin{array}{ll} \text{minimize} & D(s\|s^{(p)}) \\ \text{subject to} & Rs = t, \qquad s \succeq 0 \end{array}$$

and showed that the extended iterative method convergences to the unique optimal solution. It is interesting to note that Kruithof's method appears to be the first iterative scaling method in statistics, and that these methods are closely related to the celebrated EM-algorithm [6].

Recently [24], Zhang *et al.* have suggested to use the related criterion

$$\begin{array}{ll} \text{minimize} & \|Rs - t\|_2^2 + \sigma^{-2} D(s\|s^{(p)}) \\ \text{subject to} & s \succeq 0 \end{array} \tag{5.6}$$

for estimating traffic matrices for backbone IP traffic. The practical advantage of this formulation, which we will refer to as the Entropy approach, is that the optimization problem admits a solution even if the system of linear constraints is inconsistent. We will comment on possible choices of prior matrices at the end of this section.

Estimation under Poissonian and Generalized Linear Modeling Assumptions

Vardi [20] suggested to use a Poissonian model for the traffic, i.e., to assume that

$$s_p \sim \text{Poisson}(\lambda_p)$$

and showed that the mean and covariance matrix of the link loads are given by

$$\mathbf{E}\{t\} = R\lambda \qquad\qquad \mathbf{Cov}\{t\} = R\,\text{diag}(\lambda)\,R^T$$

A key observation is that the Poissonian model provides an explicit link between the mean and covariance matrix of the traffic. Based on a time-series of link load measurements, we compute the sample mean and covariance,

$$\hat{t} = \frac{1}{K}\sum_{k=1}^{K} t[k] \qquad\qquad \hat{\Sigma} = \frac{1}{K}\sum_{k=1}^{K}(t[k]-\hat{t})(t[k]-\hat{t})^T$$

and then match the measured moments with the theoretical, *i.e.*, solve

$$R\lambda = \hat{t} \qquad\qquad R\,\mathrm{diag}(\lambda)\,R^T = \hat{\Sigma}$$

for the vector λ of mean traffic rates. By accounting for the model of the covariance matrix we get $L(L+1)/2$ additional relations, and Vardi proves that the combined information makes the vector λ statistically identifiable. In practice, however, there will typically be no vector λ that attains equality in the moment matching conditions (this may for example be due to lack of data, outliers, or violated modeling assumptions). Vardi suggests to use the EM algorithm to minimize the Kullback-Leibler distance between the observed sample moments and their theoretical values. However, as pointed out in [5], when the observed values are not guaranteed to be non-negative, it is more reasonable to use a least squares fit. To this end, we find the estimate λ by solving the non-negative least-squares problem

$$\begin{array}{ll}\text{minimize} & \|R\lambda - \hat{t}\|_2^2 + \sigma^{-2}\|R\mathrm{diag}(\lambda)R - \hat{\Sigma}\|_2^2 \\ \text{subject to} & \lambda \succeq 0\end{array}$$

The parameter $\sigma^{-2} \in [0,1]$ reflects our faith in the Poissonian modeling assumption (compare [20, Section 4]): if σ^{-2} tends to zero, then we base our estimate solely on the first moments, while $\sigma^{-2} = 1$ is natural if we believe in the Poisson assumption.

Cao *et al.* [3] have extended the Vardi's approach by considering a generalized linear modeling assumption

$$s_p \sim \mathcal{N}\left(\lambda_p, \phi\lambda_p^c\right)$$

and assumes that all source-destination flows are independent. The additional scaling parameters ϕ and c give somewhat more freedom than the strict Poissonian assumption. However, even for fixed scaling constants ϕ and c, the estimation procedure is more complex (the associated optimization problem is non-convex), and Cao *et al.* propose a pseudo-EM method for estimation under fixed value of c. An interesting aspect of the paper by Cao *et al.* is that they also try to account for time-variations in the OD flows in order to use more measurements than the 12 link count vectors logged during a busy hour.

Regularized and Bayesian Methods

A related class of methods can be motivated from Bayesian statistics [19]. For example, by modeling our prior knowledge of the traffic matrix as

$$s \sim \mathcal{N}(s^{(p)}, \sigma^2 I)$$

and assuming that the traffic measurements are subject to white noise with unit variance, i.e.

$$t = Rs + v$$

with $\mathbf{E}\{v\} = 0$, $\mathbf{Cov}\{v\} = I$, the maximum a posteriori (MAP) estimate is found by solving

$$\text{minimize} \quad \|Rs - t\|_2^2 + \sigma^{-2}\|s - s^{(p)}\|_2^2 \tag{5.7}$$

Once again, the optimal estimate can be computed by minimizing a weighted distance of the errors between theoretical and observed means and the distance between the estimated demands and a prior "guesstimate". The variance σ^2 in the prior model is typically used as a tuning parameter to weigh the relative importance that we should put on the two criteria. The formulation (5.7) has been used in, for example, [23], where the prior is computed using a gravity model.

Fanout Estimation

Although fanout estimation does not simplify the estimation problem if we only use a single snapshot of the link loads, it can be useful when we have a time-series of link load measurements. As discussed in Section 5.3, the fanout formulation of (5.2) is the one of finding a non-negative vector $\alpha[k]$ such as

$$RS[k]\alpha[k] = t[k], \qquad \sum_m \alpha_{nm}[k] = 1, \quad n = 1, \ldots, N$$

where $S[k]$ is a diagonal scaling matrix such that $s[k] = S[k]\alpha[k]$.

Given a time series of link load measurements, we may assume that the fanouts are constant (i.e., that all link load fluctuations are due to changes in the total traffic generated by each node) and try to find $\alpha \succeq 0$ satisfying

$$\begin{aligned} RS[k]\alpha &= t[k], & k &= 1, \ldots, K, \\ \sum_m \alpha_{nm} &= 1, & n &= 1, \ldots, N \end{aligned}$$

Even if the routing matrix itself does not have full rank, the above system of equations will quickly become overdetermined, and there is a unique vector α that minimizes the errors (in a given norm) between the observed link counts and the

ones predicted by the constant-fanout model. These can be found by solving the optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{K} \|RS[k]\alpha - t[k]\|_2^2 \\ \text{subject to} & \sum_{n=1}^{N} \alpha_{nm} = 1, \qquad m = 1, \ldots, N \end{array}$$

which is simply an equality-constrained quadratic programming problem.

Deterministic Approaches

Worst-case bounds on demands

In addition to statistical estimates, it is also interesting to find upper and lower bounds on the demands. Making no underlying statistical assumptions on the demands, we note that a single measurement $t[k]$ of the link loads could be generated by the set of possible communication rates,

$$\mathcal{S} = \{s \succeq 0 \mid Rs = t[k]\}$$

Thus, an upper bound on demand p can be computed by solving the linear programming problem

$$\begin{array}{ll} \text{maximize} & s_p \\ \text{subject to} & Rs = t[k], \qquad s \succeq 0 \end{array}$$

The associated lower bound is found by minimizing s_p subject to the constraints. Obviously, this approach is only interesting when it finds an upper bound smaller than the trivial $\max_{l \in \mathcal{L}(p)} t_l[k]$ and a lower bound greater than zero. Also note that the method is computationally expensive, as it requires solving two linear programs for each point-to-point demand.

5.5 Benchmarking the Methods on Real Data

A major contribution of this paper is to study the traffic in the backbone of a commercial Internet operator, and to benchmark the existing traffic matrix estimation methods on this data. A complete traffic matrix is measured using the operator's MPLS-enabled network.

Previous work also validated estimation methods on real data, but they instead used NetFlow data to measure the traffic matrix on single router or on a partial network. [23] validates the tomogravity method with NetFlow measurements of 2/3 of a tier-1 IP backbone, using hourly traffic matrices. In [3] NetFlow data from a single router is used to create traffic matrices in 5 minute increments, for validating time-varying network tomography.

NetFlow exports flow information from the routers to a collector system. The exported information contains the start and end time of every flow, and the number of bytes transmitted during that interval. The collector calculates the average

rate during the lifetime of the flow, and adds that to the traffic matrix. For validating time-varying tomography, this is not a very accurate methodology. The variability within a flow is lost because of the NetFlow aggregation. This might affect the variance-mean relationship this method is based on.

Our study provides new results in the sense that it uses a *full* network traffic matrix, based on the *direct measurements* (rather than analysis of NetFlow traces) of all demands at *5 minute intervals*.

In the remaining parts of this section, we describe how a complete traffic matrix is measured using Global Crossing's MPLS-enabled network, investigate some basic properties of the demands, and evaluate the existing methods for traffic matrix estimation on the data.

Data Collection and Evaluation Data Set

Network

Global Crossing is using MPLS for Traffic Engineering on its global IP backbone. A mesh of Label Switched Paths (LSPs, a.k.a. "tunnels") has been established between all the core routers in the network. Every LSP has a bandwidth value associated with it, and the core router originating the LSP (head-end) will use a constraint based routing algorithm (CSPF) to find the shortest path that has the required bandwidth available. RSVP is then used to setup the actual path across the network. This architecture is described in detail in [22].

By measuring the utilization of every LSP in 5 minute intervals using SNMP, we can create a full and accurate traffic matrix of the network. This is an additional, but important, benefit of running an MPLS-enabled network.

Data Collection

To collect SNMP data from the network, a geographically distributed system of "pollers" has been set up. Each poller retrieves SNMP information from a dedicated set of routers in its area, and also functions as a backup for neighboring pollers. SNMP uses the unreliable UDP protocol for communications between the routers and monitoring systems, and hence there is the risk of losing data during transmission. A distributed system with the pollers located close to the routers being monitored increases the reliability in the case of network performance issues or outages, and keeps the load per poller manageable.

The link and LSP utilizations are collected every 5 minutes, at fixed timestamps (e.g. 9:00:00, 9:05:00, 9:10:00, etc.). There will be some variation in the exact polling time, as it is impossible to query every router and interface at exactly the same time. The exact response time of the routers is recorded, and the corresponding utilization rate data is adjusted for the length of the real measurement interval (e.g. 5 minutes and 3 seconds). The impact of this on the measurements is only minimal, and it provides uniform time series of link and LSP utilization data.

The pollers transfer their data to a central database at fixed intervals, using a reliable transport protocol (TCP).

Routing Matrix

The routing matrix in the form described by equations (5.1) and (5.2) is created using a simulation of the network. Although the routing of the LSPs in the network could be retrieved from the routers, it proves to be more practical to simulate the constraint based routing protocol (CSPF) as used by the routers, using the same constraints data (i.e. LSP bandwidth values).

We use the tool MATE from Cariden [4] to perform this routing simulation, and export this information in a text file. The data is then converted to a routing matrix according to equation (5.1). Although the routing in the network is often in a state of flux because of link and/or equipment outages, this is not of much relevance to our study. These routing changes only have a minor effect on the point-to-point demands (i.e., the traffic matrix).

Evaluation Data Set

In order to perform a scientific evaluation of the estimation methods, we need the measurements of routing, traffic matrix elements and link loads to be consistent. By consistent we mean measurements which satisfies equation (5.2).

By using equation (5.2) we are able to compute the link loads needed as input to the estimation methods, from the measured point-to-point demands and the simulated routing matrix. The above mentioned procedure enable us to evaluate the accuracy of the methods on real data without the errors incurred by errors in the measurement of the link loads.

From Global Crossing's network, we have extracted routing information and traffic matrices for the European and American subnetworks. The reason for this is that we wanted to study networks of manageable size that still accommodate large traffic demands. It also allows us to study if there are any significant differences in the demand patterns on the two continents. To create these separate traffic and routing matrices, we simply exclude all links and demands that do not have both source and destination inside the specific region.

Further, core routers located in the same city were aggregated to form a point of presence (PoP), and we study the PoP-to-PoP traffic matrix. Many PoPs contain routers who only transit traffic. We have in this study included links between these transit routers since we focus on estimation in real networks where transit routers are present. Because not necessarily all the original demands between two PoPs were following the same path, we decided to route the aggregated demand according to the routing of the largest original demand. In practice though, most parallel demands already followed the same path.

Using this approach, the European network has 12 PoPs (thus 132 point-to-point demands) and 72 links, while the American network has 25 PoPs (600 demands) and 284 links.

Since the precise details of the traffic are considered proprietary, we scale all plots by the maximum value of the total traffic during the measurement period. It might, however, be interesting to know that the largest traffic demands are on the order of 1200 Mbps.

Preliminary Data Analysis

Busy hours and demand distributions

Figure 5.1 shows how the normalized total traffic in the two subnetworks vary with time. The solid and dashed lines represent the European and American networks, respectively. There is a clear diurnal cycle, and both subnetworks have a pronounced busy periods. The busy periods overlap partly around 18:00 GMT, and the time period shaded in Figure 5.1. We will focus our data analysis to this interval.

Figure 5.2 shows the cumulative demand distribution for the subnetworks. The figure shows that the top 20 percent of demands account for approximately 80 percent of the traffic in both networks. A similar insight can be obtained from the spatial traffic distributions illustrated in Figure 5.3, where we see that a limited subset of nodes account for the majority of network traffic.

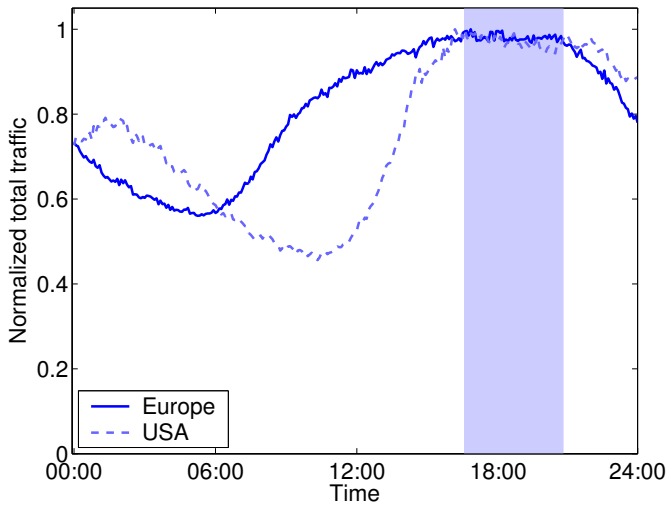

Figure 5.1: Total network traffic over time. The solid line represents the European network, while the dashed line represents the American subnetwork.

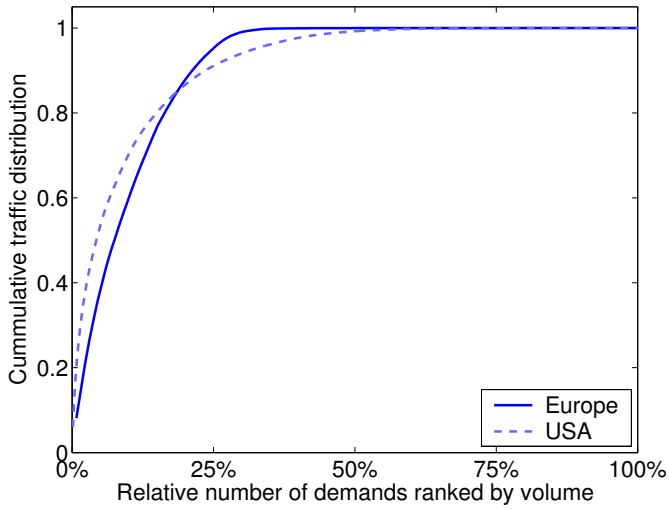

Figure 5.2: Cumulative demand distributions for the European network (solid) and the American subnetwork (dashed).

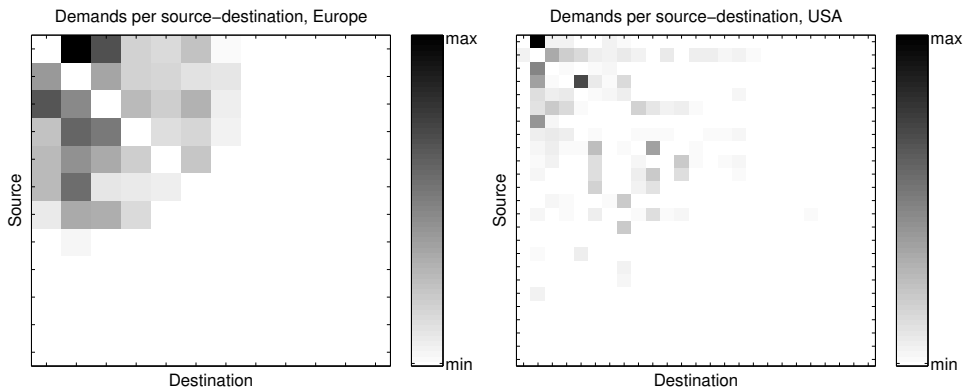

Figure 5.3: Spatial distribution of traffic in the two subnetworks.

On the stability of fanout factors

As we have seen in Section 5.3, there are several possible formulations of the traffic estimation problem: we may estimate the demands directly, focus on the relative demands (viewing the traffic matrix as a demand distribution) or the fanout factors. While the total network traffic changes with the number of active users, one may conjecture that fanouts would be stable as long as the average user behavior does not change. In this section, we investigate whether fanouts are more stable over time than demands themselves. If this is the case, fanout estimation may be easier than demand estimation since we do not have to rely on data logged only during the stationary busy hour. Furthermore, if fanouts are stable, it is a worthwhile idea to develop models for fanout factors based on node characteristics (cf. [12]).

Figure 5.4 shows how the demands from the four largest PoPs in the American network fluctuate over the 24-hour measurement period, while Figure 5.5 shows the associated fanouts. We can see that the fanouts are much more stable than the demand themselves during this measurement period. The same qualitative relationship holds for all large demands in the network; for the smaller demands, however, the fanouts sometimes fluctuate more than the demands themselves.

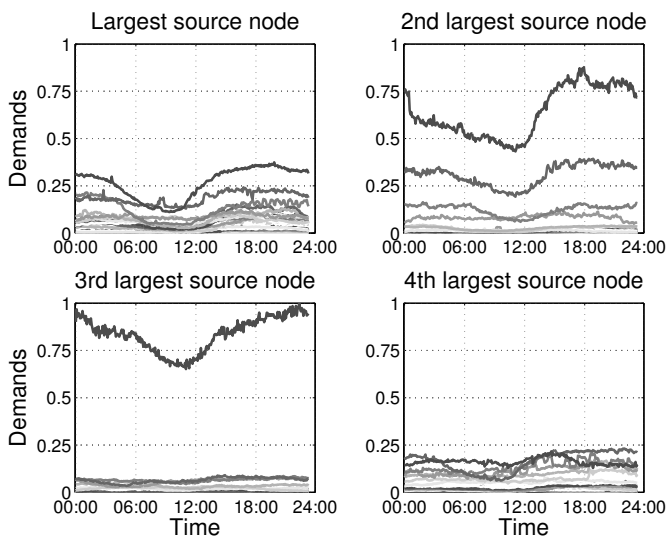

Figure 5.4: The four largest outgoing demands from the four largest PoPs in the American network.

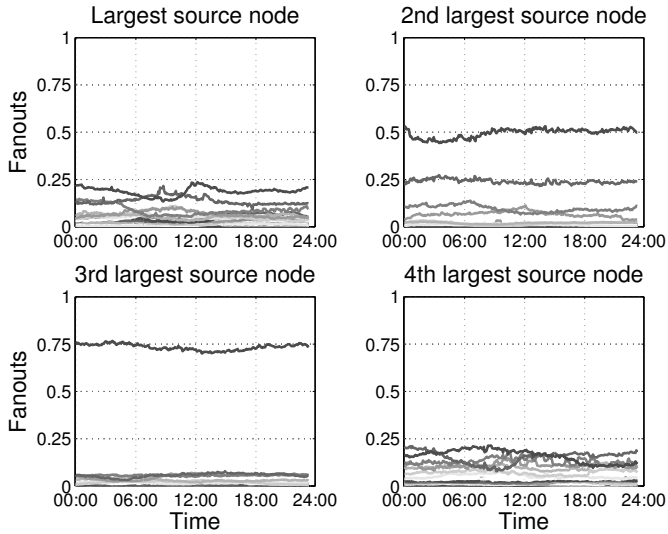

Figure 5.5: The associated fanouts for the four largest outgoing demands from the four largest PoPs in the American network.

On the Poissonian Modeling Assumption

The assumption that demands are Poissonian, or that they follow a generalized scaling law, provides an explicit link between mean and covariances of link load measurements. Such a link allows us, at least in theory, to statistically identify the demands based on a time series of link load measurements. It is therefore interesting to investigate how well our data satisfies the generalized scaling law [3]

$$\mathbf{Var}\left\{s_p\right\} = \phi\lambda_p^c$$

In particular, if the traffic is Poissonian, then $\phi = c = 1$. Figure 5.6 shows the relationship between the 5-minute averages of mean and variance for the demands in our subnetworks during busy hour. The plots show a remarkably strong relation between mean and variance and that the generalized scaling law is able to capture the mean-variance relationship for the demands in both subnetworks. The parameters $\phi = 0.82, c = 1.6$ gives the best fit for the European demands, and $\phi = 2.44, c = 1.5$ results in the best fit for the American network.

Similar mean-variance relationships have been established for web-traffic in [14] and for IP traffic demands in [3, 13]. Our observations are consistent with the measurements on a single LAN router in [3] (which suggest that $c = 2$ is more reasonable than the Poissonian assumption $c = 1$), but differs from the measurements on the Sprint backbone reported in [13] (which finds that c varies uniformly over

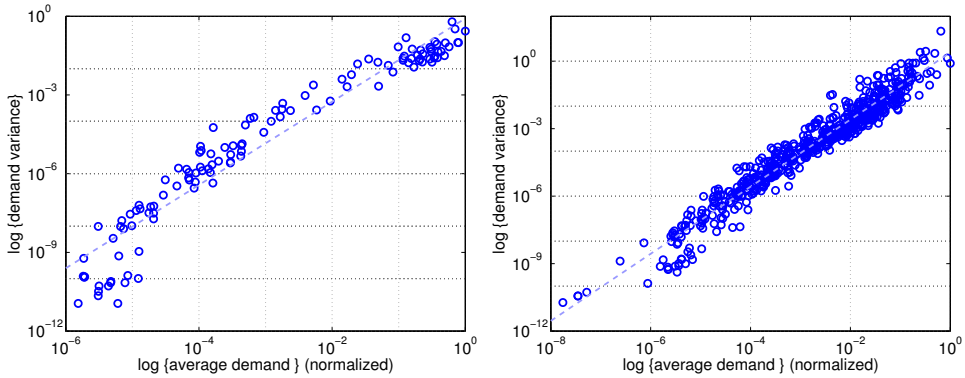

Figure 5.6: Relation between mean and variance for the demands in the European (left) and American (right) subnetworks.

the interval $[0.5, 4.0]$). This difference could be explained from the fact that [13] calculates the 1-second mean-variance relationship *per demand* over 400 intervals of 100 seconds each. The variation of the per demand mean over these 400 intervals (a little more than 11 hours) is not going to be very large. In our analysis, we use the 5-minute mean-variance numbers from all demands during a single interval, like the busy hour for which we want to estimate the traffic matrix. This way we fit the data over an average demand range of 6 magnitudes or more, based on the same measurement intervals that will be used for the estimation procedures.

On the Gravity model assumption

Finally, we investigate to what extent the gravity model provides a good estimate of the demands. We focus our analysis on the simple gravity model although the generalized gravity model potentially yield more accurate results since the latter model requires information we do not have access to. Figure 5.7 shows the actual traffic matrix elements against the gravity model estimates. While the gravity model is reasonably accurate for the European network, it significantly underestimates the large demands in the American network. With our knowledge about the spatial distribution of demands shown in Figure 5.3 we could have foreseen this result. Contrary to the gravity model assumption that all PoPs send the same fraction of their total traffic to each destination, PoPs tend to have a few dominating destinations that differ from PoP to PoP.

Evaluation of Traffic Matrix Estimation Methods

In this section, we evaluate the methods for traffic matrix estimation described in Section 5.4. Since fanout estimation and the Vardi approach both use a time-series

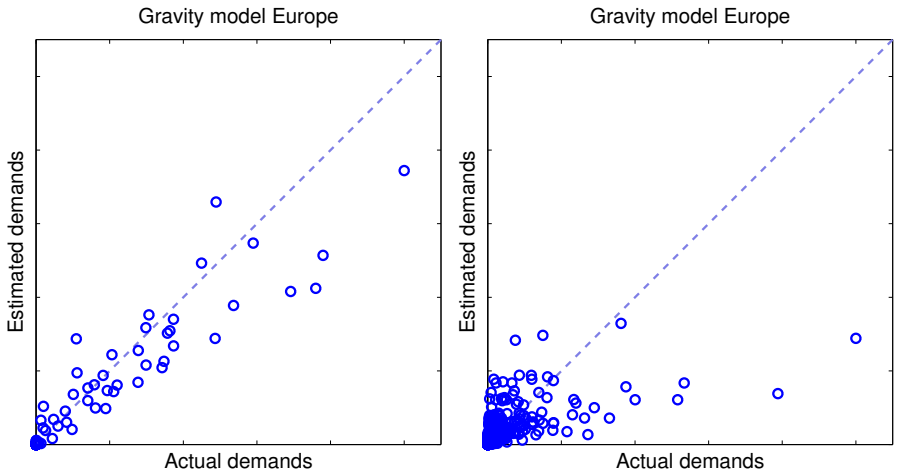

Figure 5.7: Real demands vs. gravity model estimates for European (left) and American (right) subnetworks.

of measurements rather than a snapshot, they are analyzed separately from the other methods.

Performance Metrics

To evaluate the methods, we must first determine an appropriate performance measure. Although many aspects could potentially be included in the evaluation, we focus on the potential impact of performance errors on traffic engineering tasks such as load balancing or failure analysis. For these applications, it is most important to have accurate estimation of the largest demands since the small demands have little influence on the link utilizations in the backbone. We will thus focus our performance analysis on how well the methods are able to estimate the large demands. In order to quantify performance of the estimation and compare results from different estimation methods we introduce the mean relative error (MRE):

$$MRE = \frac{1}{N_T} \sum_{i:s_i > s_T} \left| \frac{\hat{s}_i - s_i}{s_i} \right| \tag{5.8}$$

Here, s_i denotes the true traffic matrix element and $\hat{s}_i$ denotes the corresponding estimate. The sum is taken over the elements in s larger than s_T and N_T is the number of elements in s larger than the threshold. In our analysis, we have chosen the threshold so that the demands under consideration carry approximately 90% of the total traffic. This corresponds to including the 29 largest demands in the European subnetwork, and the 155 largest demands in the American network.

Evaluation of Worst-Case Bounds

To get a feel for how difficult it is to estimate different demands, it is useful to compute worst-case bounds for the demands using the approach described in Section 5.4. The resulting bounds for are shown in Figure 5.8.

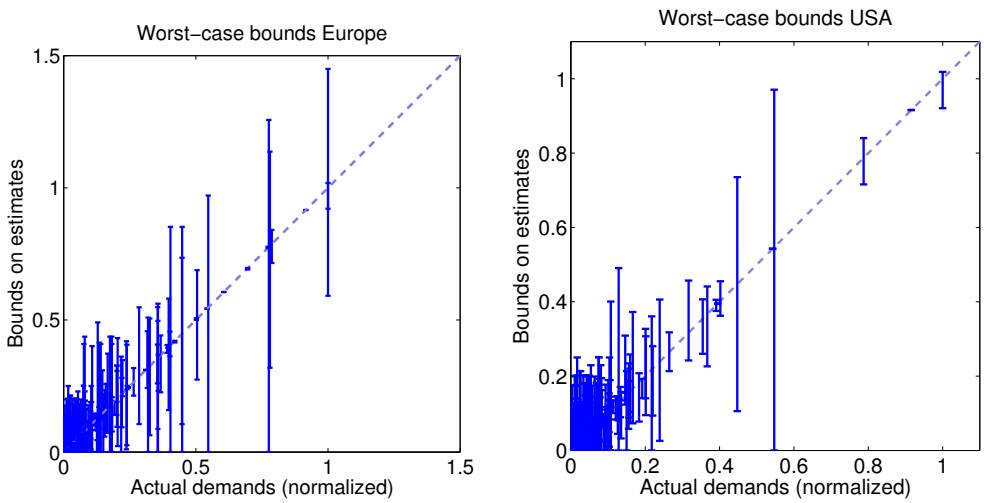

Figure 5.8: Worst-case bounds on demands in European (left) and American (right) subnetworks.

Although most bounds are non-trivial, they tend to be relatively loose and only very few bounds can be measured exactly. Still, as shown in Figure 5.9 the average of the upper and lower bound for each flow gives a relatively accurate estimate of the demands. We can observe that many of the largest demands in the European subnetwork have relatively large worst-case bounds, indicating potentially large uncertainty in the estimates.

Evaluation of Fanout Estimation

Figure 5.10 shows the results of the fanout-based estimation scheme on the American subnetwork. Since the approach uses a time-series of link load measurements, we have the average demands over the time window on the x-axis against the estimated average demand on the y-axis. Although the system of equations becomes overdetermined already for a window length of 3, the actual performance only improves marginally as we include more data.

To quantify the error we plot the MRE as a function of the window length as shown in Figure 5.11. The figure shows that the error decreases for short time-series of measurements, but levels out for larger window sizes.

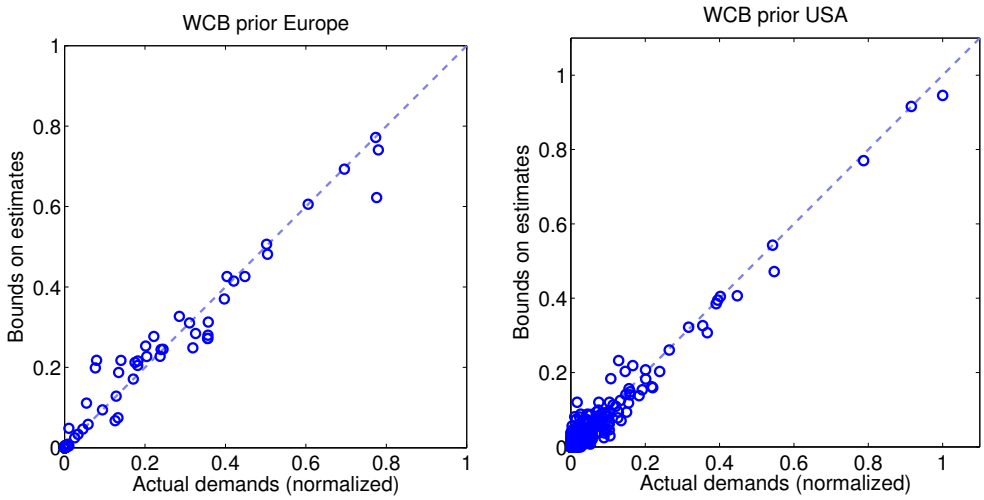

Figure 5.9: Priors obtained from worst-case bounds.

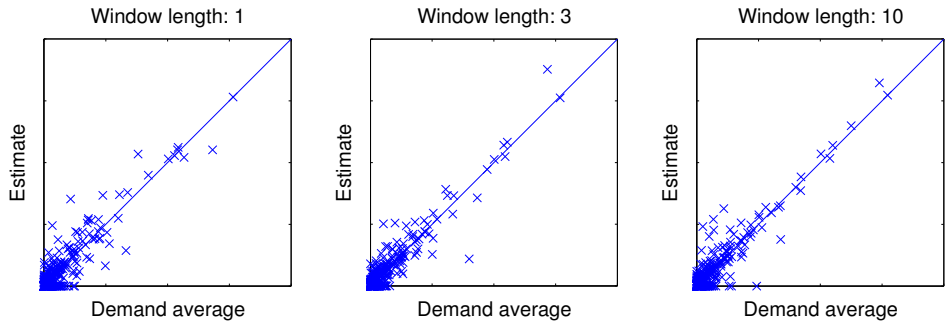

Figure 5.10: Average demands over time window vs. estimates for the fanout estimation procedure using actual data from American subnetwork.

Evaluation of the Vardi approach

In the analysis of the Vardi approach, we apply the method on the busy period of respective network (*i.e.*, the shaded interval in Figure 5.1). The busy period is 250 minutes, or 50 samples long, and we use the sample mean of the traffic demands over the busy period as the reference value in the MRE calculations.

Table 5.1 shows MRE for $\sigma^{-2} = 0.01$ and $\sigma^{-2} = 1$. The value $\sigma^{-2} = 1$, which corresponds to strong faith in the Poisson assumption, gives unacceptable performance; some estimates are several orders of magnitude larger than the true

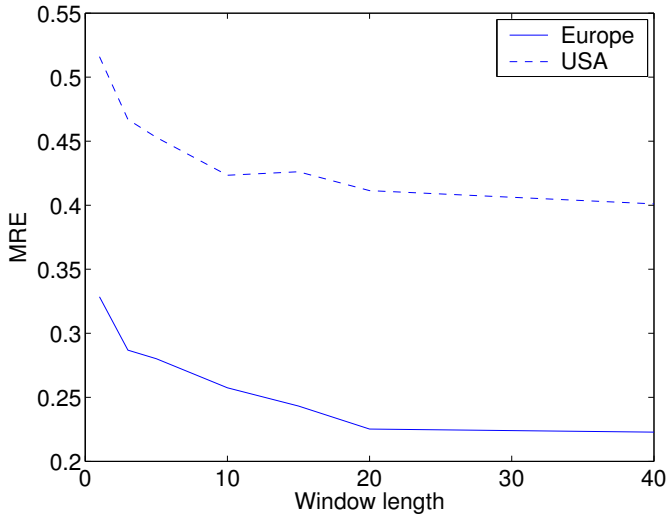

Figure 5.11: MRE as a function of window length.

	Europe	America
$\sigma^{-2} = 0.01$	0.47	0.98
$\sigma^{-2} = 1$	302	1183

Table 5.1: MRE for the Vardi approach, $K = 50$

demands while other elements are set to zero despite that the corresponding demand is non-zero. Smaller values of σ give better performance, but are still not very convincing. We believe there are two reasons for the poor performance. First, although there is a strong mean-variance relationship, the analysis in Section 5.5 has shown that the demands are not Poissonian. Second, the convergence of the covariance matrix estimation is slow and one needs a large set of samples to have an accurate estimate. To support this argument, we calculate the mean of the elements of the traffic matrix over the busy period and generate a time-series of synthetic traffic matrices with Poisson distributed elements with the calculated mean. Figure 5.12 shows MRE as a function of window size for synthetically generated traffic matrices. The solid line shows the error for the European network and the dashed line the error for the American network. To have errors in the estimation less than 20% we need a window size of 100 for the American network. Hence, even when the Poisson assumption is valid, a large window size is needed in order to achieve an acceptable level of the estimation error.

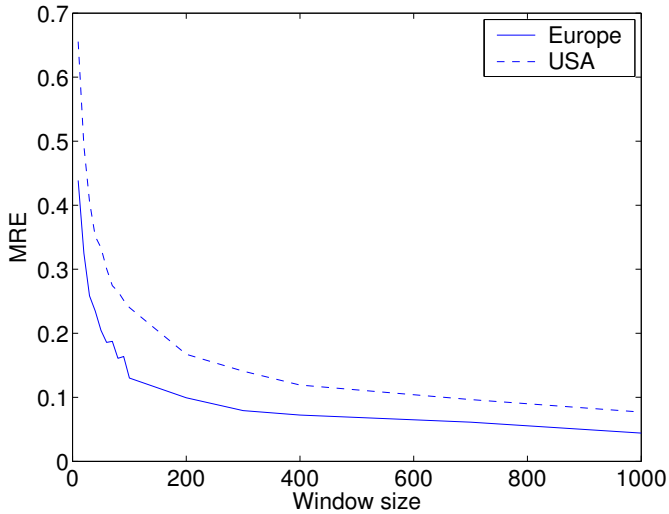

Figure 5.12: MRE as a function of window size for a synthetic traffic matrix, $\sigma^{-2} = 1$

Comparison of Bayesian and Entropy models

In this section, we evaluate the methods that use a single snapshot measurement from the network. We use the simple gravity model as prior. As before, the threshold value of the MRE method is adjusted so that approximately 90% of the total traffic in the network is included in the study.

Relying on regularization, the results of both the Bayesian (5.7) and the Entropy (5.6) approach depend on the choice of regularization parameter. For a small values of σ we make little use of the measurement and focus on finding a solution that is close to the prior. For very large values of σ, on the other hand, we put a strong emphasis on the measurements, and only use prior to select the most plausible solutions of the demand estimates that satisfy $Rs = t$. This is clearly shown in Figure 5.13, where we have computed the MRE values for both methods as function of the regularization parameter. The leftmost values should be compared with the MRE of the gravity prior, which is 0.26 in European and 0.8 in the American subnetwork. As the plots show, we get the best results for large values of the regularization parameter. We can also see that there is no single best method; the Bayesian performs better in Europe while the Entropy approach works better in the American subnetwork.

To gain intuition about the performance of the estimation we have plotted the actual traffic matrix elements against the estimated for the American network. Figure 5.14 shows the plot for Bayesian (left) and Entropy (right) estimation. The regularization parameter was set to 1000 producing the best possible estimation for

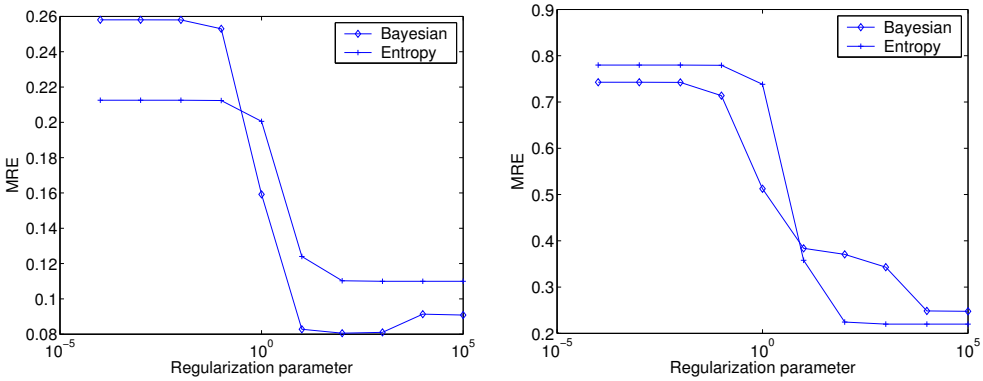

Figure 5.13: Mean relative error (MRE) as a function of the regularization parameter for the European network (left) and the American network (right).

both Bayesian and Entropy estimation. The plots show that the estimation manage to capture the traffic demands for the whole spectrum of traffic demands.

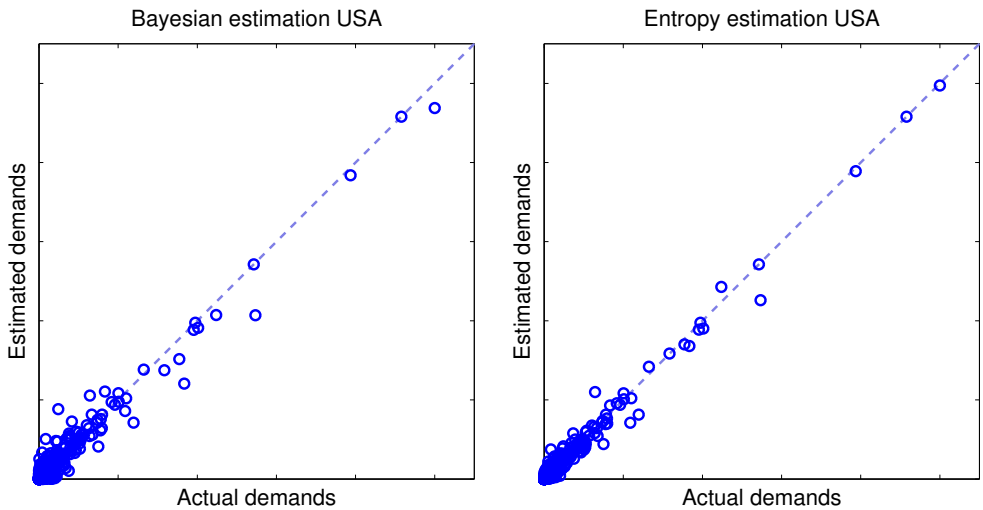

Figure 5.14: Real vs. estimated traffic demands for the American subnetwork using the Bayesian approach (left) and Entropy estimation (right).

Finally, we have demonstrated that using the mean of the upper and lower worst-case bound for each demand resulted in an estimate which is significantly better than the gravity model, and is thus natural to use this as an alternative

prior in the regularized approaches. Figure 5.15 shows the MRE for the Bayesian approach as function of regularization parameter for the gravity and worst-case bound prior on the European (left) subnetwork and the American (right) subnetwork. We can see that the worst-case bound prior gives significantly better results for small values of the regularization constant (*i.e.* when we put large emphasis on the prior). For large values of the regularization parameter, however, the performance of the two priors is practically equal.

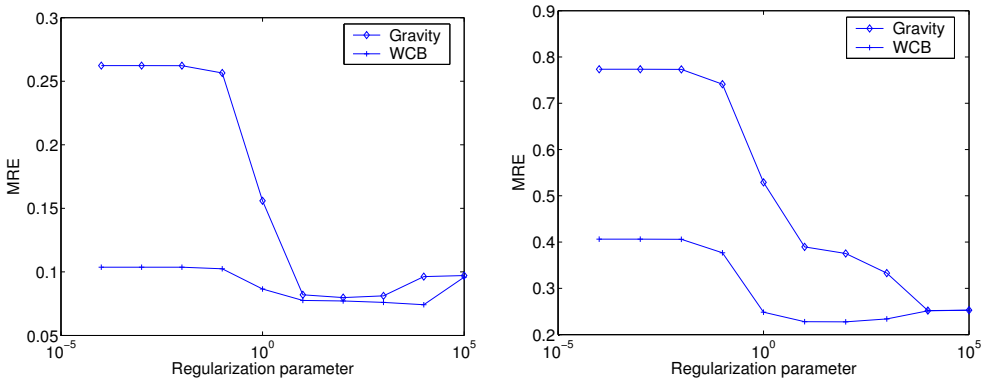

Figure 5.15: Mean relative error (MRE) as a function of the regularization parameter for the European (left) network and the American (right) network using gravity and worst-case bound priors.

Combining Tomography with Direct Measurements

As a final exercise, we investigate the usefulness of combining traffic matrix estimation based on link-loads with direct measurements of specific demands. To get correspondence with the rest of this paper, we focus on the problem of adding measurements that allow us to decrease the MRE of the Entropy method.

Figure 5.16 shows how the MRE for the Entropy approach decreases with the number of measured demands for the European subnetwork. We can see that it is sufficient to measure six demands in order for the MRE to drop from the initial 11% to below 1%. For the American network, on the other hand, we need to measure 17 demands for the MRE to decrease from the initial 23% to below 10%. These results are generated by finding, by exhaustive search in each step, the demand that when measured gives the largest decrease in MRE. They indicate that significant performance improvements can be achieved by measuring only a handful demands.

In practice, however, one would also need an approach for choosing the best demand. Comparing Figures 5.16 and 5.1, one is easily led to believe that they are

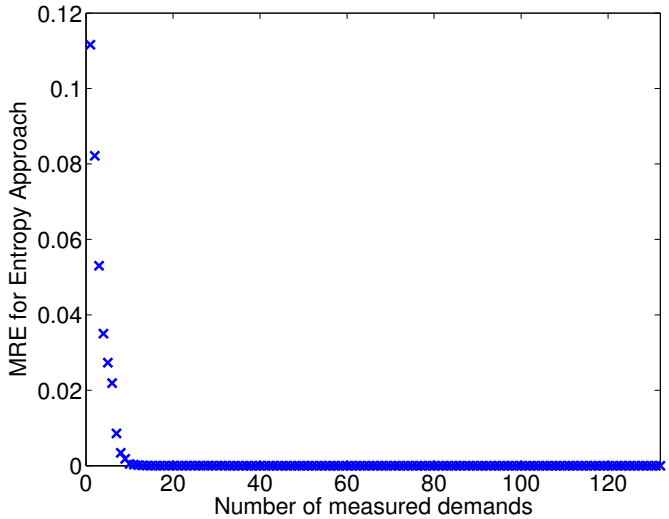

Figure 5.16: The MRE versus number of demands that we measure exactly in the European network.

nothing but each others' inverses, and it would be sufficient to measure the largest demands. In passing, we note that most estimation methods are very accurate in ranking the size of demands, so identifying the largest demands and measuring them is indeed a viable practical approach. However, the MRE measures the relative error, and in our data set, it is not the largest demands that have the largest relative estimation errors. In Europe, one would need to measure the 19 largest demands to have a MRE less than 1%, and in the American network, one would need to measure 74 demands to force the MRE below 10%.

Evaluation in summary

To allow an easy performance comparison of the methods, Table 5.2 summarizes the best MRE values that we have been able to achieve for the different approaches. The table demonstrates that the Bayesian and Entropy methods gave the best performance, followed by the fanout and Vardi approaches. The worst-case bounds provide a better prior than the simple gravity model on our dataset, and both methods provide better MRE values than the Vardi approach. Note, however, that the fanout and Vardi approaches use, and are evaluated on, a sequence of link load measurements.

Since our experiences of other aspects of the methods, such as ease-of-use and computational complexity, are not easily summarized in a single numbers, we have omitted a direct comparison and refer to the discussions above.

	Europe	America
Worst-case bound prior	0.10	0.39
Simple gravity prior	0.26	0.78
Entropy w. gravity prior	0.11	0.22
Bayes w. gravity prior	0.08	0.25
Bayes w. WCB prior	0.07	0.23
Fanout	0.22	0.40
Vardi	0.47	0.98

Table 5.2: Performance comparison of the various methods. The table shows the best MRE values that we have been able to achieve for the various methods on the two subnetworks.

5.6 Conclusion and Future Work

This paper has presented an evaluation of traffic matrix estimation techniques on data from a large IP backbone. In contrast to previous studies that used partial traffic matrices or demands estimated from aggregated NetFlow traces, we have used a unique data set of complete traffic matrices measured over five-minute intervals. The data set has allowed us to do accurate data analysis on the time-scale of standard link-load measurements and enabled us to evaluate both methods that use a time-series of link-loads and methods that rely on snapshot measurements.

We have shown that the demands in our data set have a remarkably strong mean-variance relationship, yet we have been unable to achieve good estimation performance using methods that try to exploit this fact. We have argued that this failure is due the problem of accurate estimation of covariance matrices and presented a study on synthetic data to support this claim.

Based on our observation that fanout factors tend to be much more stable over time than the demands themselves, we have proposed a novel method for estimating fanouts based on a time-series of link load measurements. We have also proposed to estimate worst-case bounds on the demands. Although these bounds are not always very tight, they turned out to be useful for constructing a prior for use in other estimation schemes. We have illustrated that the gravity model fails to construct a good prior in one of our subnetworks due to violations of underlying assumptions in the traffic patterns. The regularized methods, such as Bayesian and Entropy approaches, were found to be simple and provide the best results, if the regularization parameter was chosen appropriately. Finally, we noted that by measuring only a handful of demands directly, it was possible to obtain significant decreases in the MRE of the Entropy approach.

This study has focused on analyzing key properties of the demand data set and evaluating the performance of traffic matrix estimation techniques in terms of their estimation error. Although we have covered most methods from the literature, we have not implemented and evaluated the approach by Cao *et al.* [3]. Clearly, a

more complete evaluation should include also this method. It would also be useful to complement the evaluation by a more rigorous theoretical analysis to bring a better understanding of our observations. Our study also leaves many important issues unexplored. For example, our data set does not contain measurement errors or component failures and we have not evaluated the effect of such events on the estimation. Furthermore, we have not considered how sensitive traffic engineering tasks are to estimation errors in different demands, and how such information could be incorporated in the estimation procedures. Another interesting topic for future work would be to understand the nature of the worst-case bounds, and see if they could be exploited in other ways.

Acknowledgments

This work is supported in part by the Swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR), the Swedish Agency for Innovation Systems (VINNOVA) and the European Commission.

Bibliography

[1] H. Abrahamsson, J. Alonso, B. Ahlgren, A. Andersson, and P. Kreuger. A multi path routing algorithm for IP networks based on flow optimisation. In B. Stiller, M. Smirnow, M. Karsten, and P. Reichl, editors, *From QoS Provisioning to QoS Charging – Third COST 263 International Workshop on Quality of Future Internet Services, QoFIS 2002 and Second Interntational Workshop on Internet Charging and QoS Technologies, ICQT 2002*, pages 135–144, Zürich, Switzerland, October 2002. Springer. LNCS 2511.

[2] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[3] J. Cao, D. Davis, S. Vander Wiel, and B. Yu. Time-varying network tomography: router link data. *Journal of Americal Statistical Association*, 95:1063–1075, 2000.

[4] Cariden, Inc., Mountain View, CA. *MATE*, 2004. http://www.cariden.com.

[5] I. Csiszár. Why least squares and maximum entropy? – an axiomatic approach to inverse problems. *The Annals of Statistics*, 19:2033–2066, December 1991.

[6] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions, Suppl. 1*, Supplement Issue No. 1:205–237, 1984.

[7] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: Methodology and experience. In *Proc. ACM SIGCOMM*, Stockholm, Sweden, August 2000.

[8] J. Kowalski and B. Warfield. Modeling traffic demand between nodes in a telecommunications network. In *Australian Telecommunications and Networks Conference*, Sydney, Australia, December 1995.

[9] J. Kruithof. Telefoonverkeersrekening. *De Ingenieur*, 52(8):E15–E25, 1937.

[10] R. S. Krupp. Properties of Kruithof's projection method. *The Bell System Technical Journal*, 58(2):517–538, February 1979.

[11] A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya, and C. Diot. A Taxonomy of IP Traffic Matrices. In *SPIE ITCOM: Scalability and Traffic Control in IP Networks II*, Boston, August 2002.

[12] A. Medina, K. Salamatian, N. Taft, I. Matta, Y. Tsang, and C. Diot. On the convergence of statistical techniques for inferring network traffic demands. Technical Report BUCS-2003-003, Boston University, Computer Science, USA, February 2003.

[13] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *Proc. ACM SIGCOMM*, Pittsburg, USA, August 2002.

[14] R. Morris and D. Lin. Variance of aggregated web traffic. In *Proc. IEEE INFOCOM*, pages 360–366, Tel Aviv, Israel, March 2000.

[15] A. Nucci, R. Cruz, N. Taft, and C. Diot. Design of IGP link weight changes for estimation of traffic matrices. In *Proc. IEEE INFOCOM*, Hong Kong, March 2004.

[16] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in modeling backbone traffic variability: models, metrics, measurements and meaning. In *Proc. ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[17] M. Roughan, Mikkel Thorup, and Yin Zhang. Traffic engineering with estimated traffic matrices. In *Proc. ACM Internet Measurement Conference*, Miami Beach, Florida, USA, October 2003.

[18] A. Sridarhan, R. Guerin, and C. Diot. Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks. In *Proc. of IEEE INFOCOM 2003*, San Francisco, USA , November 2003.

[19] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):557–576, June 1998.

[20] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the Americal Statistical Association*, 91(433):365–377, March 1996.

[21] S. Vaton and A. Gravey. Network tomography : an iterative bayesian analysis. In *Proc. ITC 18*, Berlin, Germany, August 2003.

[22] X. Xiao, A. Hannan, B. Bailey, and L. M. Ni. Traffic engineering with MPLS in the internet. *IEEE Network*, 14(2):28–33, March–April 2000.

[23] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proc. ACM Sigmetrics*, San Diego, CA, June 2003.

[24] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[25] G. K. Zipf. Somde determinants of the circulation of information. *American Journal of Psychology*, 59:401–421, 1946.

Chapter 6

Paper B: Performance of Traffic Engineering in Operational IP-networks - an Experimental Study

Anders Gunnar, Henrik Abrahamsson, and Mattias Söderqvist. In *Proceedings of the 5th IEEE International Workshop on IP Operations and Management IPOM 2005*, Barcelona, Spain.

Abstract

Today, the main alternative for intra-domain traffic engineering in IP networks is to use different methods for setting the weights (and so decide upon the shortest-paths) in the routing protocols OSPF and IS-IS. In this paper we study how traffic engineering perform in real networks. We analyse different weight-setting methods and compare performance with the optimal solution given by a multi-commodity flow optimization problem. Further, we investigate their robustness in terms of how well they manage to cope with estimated traffic matrix data. For the evaluation we have access to network topology and traffic data from an operational IP network.

6.1 Introduction

For a network operator it is important to tune the network in order to accommodate more traffic and meet service level agreements (SLAs) made with their customers. In addition, as new bandwidth demanding and also delay and loss sensitive services are introduced it will be even more important for the operator to manage the traffic situation in the network. This process of managing the traffic is often referred to as *traffic engineering*. The aim is to use the network resources as efficiently as possible and to avoid congestion; *i.e.* deviate traffic from highly utilized links to less utilized links.

In this paper we investigate performance of traffic engineering in operational networks. To make the investigation more balanced we use two traffic engineering methods. The results are compared to the optimal routing obtained from multi commodity flow optimization and the inverse capacity weight setting recommended by Cisco. In order to optimize the routing an estimate of the traffic situation in the network is needed. The traffic situation can be captured in a traffic matrix. The entries in the traffic matrix represent the amount of traffic sent between each source destination pair in the network. However, since routers often lack functionality to measure the traffic matrix directly operators are forced to estimate it from other available data. We use two well known traffic matrix estimation methods to investigate how traffic engineering perform when subjected to estimated traffic matrices.

For the evaluation we have access to a full traffic matrix as well as network topology obtained from direct measurements in a commercial IP network. Previous work have shown that traffic engineering enables the network operator to accommodate substantially more traffic in the network. However, the evaluations have been performed on synthetic data or only partial traffic matrices obtained from an operational IP network.

Our focus is not on the actual methods we use in the study but in how they perform in a real network with real traffic demands. In addition, we study the interplay between traffic estimation and an application of the estimate, *i.e.* traffic engineering. Hence, we only give a brief description of the methods we use and the interested reader should consult the references for further details.

The rest of the paper is organized as follows. Section 6.2 give a short description of traffic engineering in IP networks. We also discuss related work on the subject. The experiments together with a short description of traffic matrix estimation is given in Section 6.3. The evaluation is described in Section 6.4. Finally we make some concluding remarks about our findings and discuss future work.

6.2 Traffic Engineering in IP Networks

Traffic engineering encompasses performance evaluation and performance optimization of operational networks. An important goal is to avoid congestion in the

network and to make better use of available network resources by adapting the routing to the current traffic situation.

The two most common intra-domain routing protocols today are OSPF (Open Shortest Path First) and IS-IS (Intermediate System to Intermediate System). They are both link-state protocols and the routing decisions are based on link costs and a shortest (least-cost) path calculation. With the equal-cost multi-path (ECMP) extension to the routing protocols the traffic can also be distributed over several paths that have the same cost.

These routing protocols are designed to be simple and robust rather than to optimize the resource usage. They do not by themselves consider network utilization and do not always make good use of network resources. The traffic is routed on the shortest path through the network even if the shortest path is overloaded and there exist alternative paths. It is up to the operator to find a set of link costs (weights) that is best suited for the current traffic situation and avoids congestion in the network.

The traffic engineering process is illustrated in Figure 6.1. The first step is to collect the necessary information about network topology and the current traffic situation. Most traffic engineering methods need as input a traffic matrix describing the demand between each pair of nodes in the network. But today the support in routers for measuring the traffic matrix is limited. Instead, an often suggested approach is to estimate the traffic matrix from link loads and routing information [5, 6, 13]. Link loads are readily obtained using the Simple Network Management Protocol (SNMP) and routing information is available from OSPF or IS-IS link-state updates.

The traffic matrix is then used as input to the routing optimization step, and the optimized parameters are finally used to update the current routing. In this study this means that the traffic matrix is used together with heuristic search methods to find the best set of links weights.

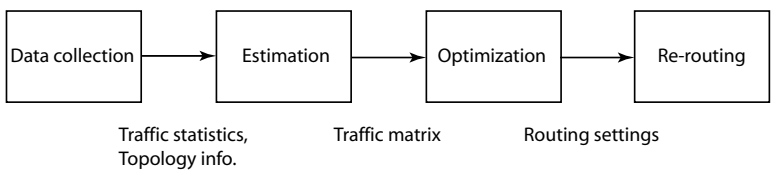

Figure 6.1: The traffic engineering process

Optimal Routing

The general problem of finding the best way to route traffic through a network can be mathematically formulated as a multi-commodity flow (MCF) optimization problem (see, e.g., [1, 3, 7]). The network is then modeled as a graph. The problem consists of routing the traffic, given by a demand matrix, in the graph

with given link capacities while minimizing a cost function. This can be formulated and solved as a linear program.

How the traffic is distributed in the network very much depends on the objectives expressed in the cost function. Since one of the main purposes with traffic engineering is to avoid congestion a reasonable objective would be to minimize the maximum link utilization in the network. Another often proposed objective function is described by Fortz and Thorup [3]. Here the sum of the cost over all links is considered and a piece-wise linear increasing cost function is applied to the flow on each link. The basic idea is that the cost should depend on the utilization of a link and that it should be cheap to use a link with small utilization while using a link that approaches 100% utilization should be heavily penalized. The characteristics of the cost function is shown in Figure 6.2.

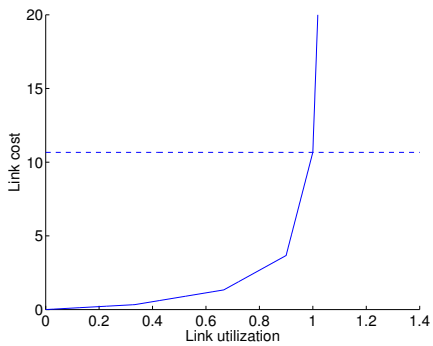

Figure 6.2: Cost function for load on a link

Though the solution given by the linear program is the optimal routing the method is in general not used directly for routing in operational IP networks. First, the method is inherently centralized. And also, since the solution requires flows to be arbitrary split among several paths towards the destination it would require modifications to the forwarding mechanisms that is used today [1]. In this paper we focus on the legacy routing protocols OSPF and IS-IS. The optimal routing will be used for comparison only since it constitutes a lower bound for the performance of legacy routing mechanisms.

Unfortunately, when taking the restrictions of shortest-paths or equal-cost multipaths in the OSPF and IS-IS protocols into consideration, the problem of finding the optimal routing becomes much harder. The problem of finding weights that optimizes the routing is NP-hard [3, 7]. This means that one usually has to rely on heuristic methods to find the set of weights.

Heuristic Search Methods

An often proposed method to determine the best set of link weights is to use local search heuristics [3, 4, 8]. Given network topology, link capacities and the demand matrix the heuristics evaluate points in a search space, where a point is represented by a set of weights. A neighbor to a point is another set of weights produced by changing the value of one or more weights from the first point. In the heuristics, different neighbors are produced and the cost of each one is calculated using a cost function. From each heuristic the neighbor with the best cost is the one that will be the output. In this study we have selected two heuristics:

- Local search (Fortz and Thorup [3])
- Strictly descending search (Ramakrishnan and Rodrigues [8])

Both heuristics have been studied for random topologies and synthetic traffic demands by Söderqvist [10]. As objective function we use the cost function by Fortz and Thorup [3] mentioned in section 6.2.

Related Work

With the prospect of better utilizing available network resources and optimizing traffic performance, a lot of research has been done in the area of traffic engineering. The general principles and requirements for traffic engineering are described in RFC 3272 [2] produced by the IETF Internet Traffic Engineering working group.

Many researchers use multi-commodity flow models for traffic engineering. The book by Pióro and Medhi [7] gives a comprehensive description of design models and optimizations methods for communication networks, including networks with shortest-path routing.

The performance of weight-setting methods using search heuristics has been investigated with real network topologies and synthetic data or partial traffic matrices in [3, 4, 9, 11]. Fortz and Thorup [3] evaluate their search heuristic using a proposed AT&T backbone network and demands projected from measurements. Sridharan *et al.* [11] use a heuristic to allocate routing prefixes to equal-cost multipaths and evaluate this using data from the Sprint backbone network. An alternative approach is to use the dual of a linear program to find a weight setting [12].

Roughan *et al.* [9] investigate the performance of traffic engineering methods with estimated traffic matrices. However, the authors use partial traffic matrices and one weight setting method only.

6.3 Methodology

This section describes the methodology in this study. First we describe the performance metrics for the experiments followed by a discussion on how the exper-

iments are conducted. Finally we give a short introduction to the traffic matrix estimation methods used in this paper.

Evaluation Metrics

Since the main objective of traffic engineering is to avoid congestion one natural metric of performance is maximum link utilization in the network. The utilization u_a of link a is defined as :

$$u_a = \frac{l_a}{c_a} \tag{6.1}$$

where the load on link a is denoted l_a and c_a is the capacity of the link. However, maximum link utilization only reflect one link in the network. To quantify performance for the routing where the whole network is taken into account we define the normalized cost:

$$\Phi^* = \frac{\Phi}{\Phi_{norm}}. \tag{6.2}$$

Here Φ is the cost function from Section 6.2 and Φ_{norm} is a normalization factor such that the normalized cost is comparable between different network topologies. Further details about the normalized cost can be found in [10].

Experimental Setup

In this paper we use an experimental approach to address the problem. We use a unique data set of complete traffic matrices and topology from a commercial IP network operator to simulate the effects of different weight settings for OSPF/IS-IS routing.

A measured traffic matrix, *i.e.* a traffic matrix without errors, together with the network topology is provided as input to the weight optimization. The output from the optimization is a new set of weights which we use to calculate the new routing. Finally, the measured traffic matrix is applied to the new routing in order to determine link utilization and calculate the normalized cost.

To obtain an estimated traffic matrix we simulate the routing with inverse capacity routing. The link loads obtained by applying the measured traffic matrix is then used to find an estimate of the traffic matrix. The estimated traffic matrix is used as input to the weight optimization algorithm. Finally, the optimized links weights are used to calculate the links loads by applying the original measured traffic matrix.

The optimal solution to the routing problem discussed in section 6.2 will serve as a benchmark for our experiments with the search heuristics. In addition, the routing from the inverse capacity weight setting is also included for comparison as it is often used by network operators and is the recommended weight setting by Cisco [3].

Traffic Matrix Estimation

The traffic matrix estimation problem has been addressed by many researchers before (*e.g.* [5, 6, 13]). In this study we focus on two estimation methods.

- Simple Gravity method
- Entropy method

The simple gravity method is based on the assumption that traffic between source node s and destination node d is proportional to the total amount of traffic sent by s and total amount of traffic destined to d. The strength of this method lies in its simplicity. However, the method is also known to be unaccurate in some situations [5].

A different approach to obtain the traffic matrix is to estimate it from link loads and routing information [6, 13]. Link loads are readily obtained using SNMP and routing is available from OSPF or IS-IS link-state updates. This approach often leads to an ill-posed estimation problem since operational IP networks typically have many more node pairs (entries in the traffic matrix) than links. In order to add more constraints to the problem additional information must be added. This information is usually in the form of some assumption made about the traffic matrix. The entropy method [13] minimizes the Kullback-Leibler distance between the estimate and a prior guess of the traffic demands. With the entropy method it is possible to obtain an accurate estimate of the traffic matrix (cf. [5, 13]). In our experiments we use the gravity method to produce a prior.

By choosing one accurate and one less accurate method we are able to make a more balanced evaluation of how estimation errors influence the performance of traffic engineering subjected to estimated traffic demands.

6.4 Results

In this section we present the results obtained from our experiments. For the evaluation we used network topologies and traffic matrices obtained from a global MPLS-enabled IP network. From the data we isolated the European and the American subnetworks in order to obtain networks of manageable size but still carry large traffic demands. In addition, we obtain two networks with slightly different characteristics. More details about the networks and traffic demands can be found in Gunnar *et al.* [5]. However, it might be interesting to mention that the European network has 12 nodes and 40 links and the American network has 25 nodes and 112 links.

As previously mentioned we use two measures of performance, the normalized cost function introduced by Fortz and Thorup [3] and maximum link utilization in the network. The results are plotted for the following methods:

- **Opt**, the optimal solution to the general routing problem. Included for comparison since it is a lower bound for the other methods.

- **InvCap**, sets the weight inversely proportional to the capacity of the link. Like Opt this method is included as a benchmark as it is the default setting recommended by Cisco.
- **FT**, the search heuristic proposed by Fortz and Thorup [3] starting from a random weight setting.
- **RR**, the search heuristic proposed by Ramakrishnan and Rodrigues [8].

For each topology the algorithms were run with different scaling on the traffic demands. The scalings were obtained by multiplying the traffic demand matrix with a scalar. All algorithms except InvCap use different weight settings for different scalings. In the results both cost and max utilization are presented for each topology and for each scaling. For all algorithms except OPT the cost and the max utilization is computed using the same weight setting. But for OPT the cost and the max utilization are computed independently, using different objective functions.

Experiments with Measured Traffic Matrices

Figure 6.3 shows the normalized cost and maximum link utilization for the American network and for both search heuristics. The plot shows that both search heuristics are close to the optimal routing given by the linear programming model. However, in the European network the results are somewhat different as Figure 6.4 reveals. Both heuristics improve performance compared to inverse capacity weight setting but neither of them are close to the optimal routing.

In comparison with previous studies we see that our findings confirm the results of Fortz and Thorup [3] who use a real network topology and a partial traffic matrix derived from Netflow measurements as well as synthetic data. Söderqvist [10] use synthetic topologies and traffic demands with power-law properties to show that optimizing weights improve network performance considerably compared to inverse capacity weight setting.

Optimizing Weights Using Estimated Traffic Demands

A somewhat controversial assumption made in the previous section is that an exact measure of the traffic matrix is available. In this section we investigate how the search heuristics perform when they are subjected to estimated traffic demands.

We focus on two well known estimation methods. The gravity method and the entropy method. Both methods have been evaluated on the data set we use in this study [5]. The simple gravity methods was shown to give a surprisingly accurate estimate in the European network despite its simplicity. In the American network, on the other hand, the gravity methods failed to give an accurate estimate of the traffic demands due to violation of the gravity assumption. The more sophisticated entropy method produced an accurate estimate for both the European and the American networks.

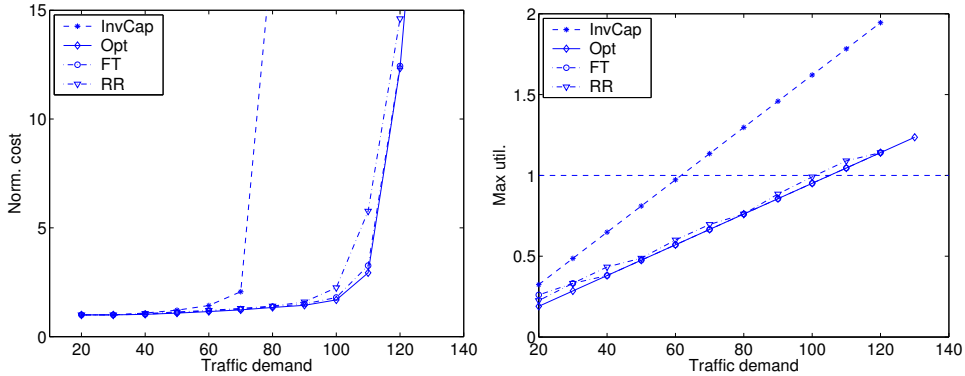

Figure 6.3: Normalized cost (left) and maximum link utilization (right) for the American network

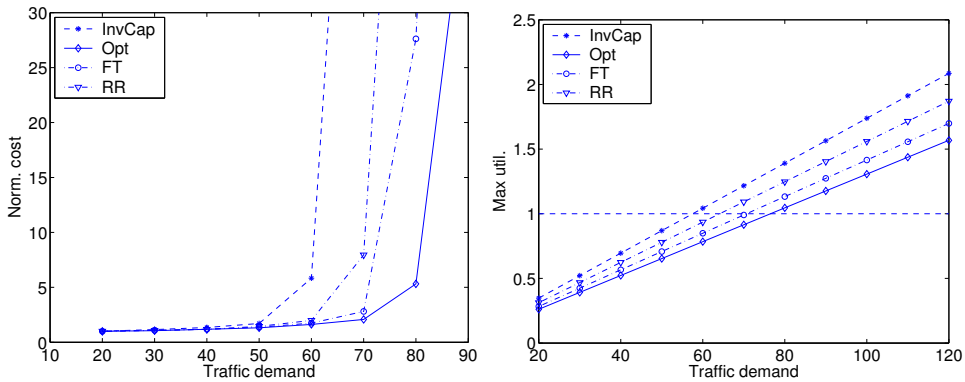

Figure 6.4: Normalized cost (left) and maximum link utilization (right) for the European network

In the plot to the left in Figure 6.5 we have plotted normalized cost as a function of traffic demands for the local search heuristic in the American network. The plot indicates that estimation using the more advanced entropy method has a negligible effect on performance. However, when the optimization is based on the less accurate gravity model performance is degraded considerably. In the European network (Figure 6.5 right), where the gravity model is more accurate, the heuristics have similar performance. The same experiment has been conducted using descending search producing similar results as local search. But we have omitted the plots for descending search due to space limitations.

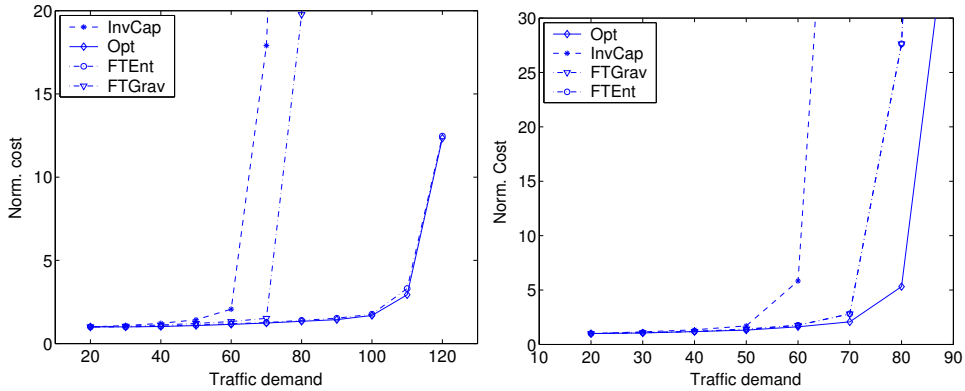

Figure 6.5: Normalized cost for the American network (left) and the European network (right) using estimated traffic matrices

6.5 Conclusions and Future Work

This paper has investigated how traffic engineering performs in a real network with real traffic demands from a commercial IP network operator. The traffic engineering methods are based on search heuristics for weight settings in link state routing. From the study we concluded that both search heuristics are able to find weight settings which are able to accommodate substantially more traffic in the network than the default inverse capacity weight setting and come close in performance to the optimal solution of the routing problem. In addition, we study the performance of the traffic engineering using estimated traffic demands. We investigate two traffic matrix estimation methods. One simple and one which is more sophisticated and accurate. Our observations indicate that when the optimized weight setting using the estimated traffic matrix from the accurate entropy method is applied to the real traffic demands performance is only degraded marginally. But for the less accurate gravity model performance was degraded significantly in some cases. However, still an improvement compared to the inverse capacity weight setting recommended by Cisco.

Our findings confirm the results form previous studies using partial traffic demands derived from flow measurements or synthetic data [3, 9, 10].

This study has focused on a static traffic matrix. In the future we intend to investigate how the weight setting can be designed to be robust in order to cope with a changing traffic situation in the network.

Acknowledgments

This paper describes work undertaken in the context of the Ambient Networks - Information Society Technologies project, which is partially funded by the Commission of the European Union. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the Ambient Networks Project.

Bibliography

[1] H. Abrahamsson, J. Alonso, B. Ahlgren, A. Andersson, and P. Kreuger. A multi path routing algorithm for IP networks based on flow optimisation. In *Proc. Third COST 263 International Workshop on Quality of Future Internet Services, QoFIS 2002*, pages 135–144, Zürich, Switzerland, October 2002. Springer. LNCS 2511.

[2] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and principles of Internet Traffic Engineering. Internet RFC 3272, May 2002.

[3] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proc. IEEE INFOCOM*, pages 519–528, Tel-Aviv, Israel, March 2000.

[4] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, May 2002.

[5] A. Gunnar, M. Johansson, and T. Telkamp. Traffic matrix estimation on a global IP backbone - a comparison on real data. In *Proc. ACM SIGCOMM Internet Measurement Conference*, pages 149–160, Taormina, Italy, October 2004.

[6] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *Proc. ACM SIGCOMM*, pages 161–174, Pittsburgh, Pennsylvania, USA, August 2002.

[7] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Commmunication and Computer Networks*. Morgan Kaufmann, 2004.

[8] K.G. Ramakrishnan and M.A. Rodrigues. Optimal routing in shortest path data networks. *Lucent Bell Labs Technical Journal*, 6(1), 2001.

[9] M. Roughan, Mikkel Thorup, and Yin Zhang. Traffic engineering with estimated traffic matrices. In *Proc. ACM Internet Measurement Conference*, pages 248–258, Miami Beach, Florida, USA, October 2003.

[10] M. Söderqvist. Search Heuristics for Load Balancing in IP-networks. Technical Report T2005:04, SICS – Swedish Institute of Computer Science, March 2005.

[11] A. Sridarhan, R. Guerin, and C. Diot. Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks. In *Proc. IEEE INFOCOM*, San Francisco, Califonia, USA , November 2003.

[12] Y. Wang, Z. Wang, and L. Zhang. Internet traffic engineering without full mesh overlaying. In *Proc. IEEE INFOCOM*, pages 565–571, Anchorage, Alaska, USA, May 2001.

[13] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information theoretic approach to traffic matrix estimation. In *Proc. ACM SIGCOMM*, pages 301–312, Karlsruhe, Germany, August 2003.

Chapter 7

Paper C: Robust Routing Under BGP Reroutes

Anders Gunnar, and Mikael Johansson. In *Proceedings of IEEE Globecom 2007*, Washington, DC, USA.

Abstract

Configuration of the routing is critical for the quality and reliability of the communication in a large IP backbone. Large traffic shifts can occur due to changes in the Inter-domain routing that are hard to control by the network operator. This paper describes a framework for modeling potential traffic shifts due to BGP reroutes, calculating worst-case traffic scenarios, and finding a single routing configuration that is robust against all possible traffic shifts due to BGP reroutes. The benefit of our approach is illustrated using BGP routing updates and network topology from an operational IP network. Experiments demonstrate that the robust routing is able to obtain a consistently strong performance under large Inter-domain routing changes.

7.1 Introduction

An important part of provisioning communication services in an IP network is managing the traffic situation. A thorough understanding of the dynamics of the traffic is necessary in order to optimize utilization of available resources and to meet service level agreements made with customers. In addition, the transfer of critical services such as telephony to IP networks has made it even more important for a network operator to monitor and control the traffic.

However, the traffic situation is highly dependent on the interplay between intra- and inter-operator routing. An operator who acts as a provider for other network operators often receives reachability information for a network from several different places. This reachability information is given in the form of a network prefix which represent the address of the network. Routing is performed by matching the destination address with the prefixes in the routing table and selecting the route with the longest prefix match. When there are several routes available a router has to select one of these routes; *i.e.* the ingress router of the traffic has to select which route to use for forwarding the traffic towards the destination network. How the selection is performed has implications on how the traffic is routed within the network since the ingress router selects an egress router where the traffic leaves the operator's network. This selection of routes may cause large shifts in the load in the network, see *e.g.* [9].

In this paper we introduce a method to control and minimize the implications of load shifts caused by changes in the inter-domain routing. In particular, we model the uncertainty of traffic demands due to BGP reroutes, formulate and solve a convex optimization problem to identify the worst-case scenarios for a given MPLS routing, and sequentially improve the routing by introducing additional tunnels that allows to hedge against these scenarios. To reduce the number of variables in the problem we devise an algorithm that identifies the prefixes with multiple egress points and large traffic volumes. In addition, worst-case scenarios are generated by considering one link at a time (finding the ingress/egress traffic demands that maximizes the utilization of each individual link, and selecting the traffic scenario that gave the largest link utilization) which allows that part of the algorithm to be highly parallelized.

Our method is applied to traffic data and inter domain routing information from an operational Internet Service Provider. We find that significant improvements are possible under a number of scenarios. For comparison we also include shortest path routing according to the original link weights as well as multi-commodity flow optimization for the nominal traffic situation in our analysis.

Optimization over multiple traffic scenarios has received a lot of attention from researchers (cf. [1, 2, 3, 6, 12]). In a pioneering paper by Fortz and Thorup [3] the authors use a search heuristic to optimize the routing over a set of traffic scenarios. Applegate and Cohen [1] calculate an upper bound for the performance of the routing under all possible traffic scenarios. The upper bound on performance is used by Wang *et al.* [12] for comparison with their method which embeds a traffic

scenario in a traffic envelope and optimizes the routing for the traffic scenario and limits performance of the routing for every traffic scenario in the envelope. In this paper we follow the approach taken by Ben-Ameur and Kerivin [2] by using column-generation to optimize the routing. However, our approach differs from previous work by incorporating inter domain routing in the solution and thereby make our results directly applicable for large IP networks with several peering points with other operators.

The rest of the paper is organized as follows. In the next section we give a short description of how routing is performed in the Internet. Section 7.3 introduces the algorithm, including the generation of worst-case traffic scenarios and the robust routing optimization. The analysis of traffic data from an operational IP network is presented in section 7.4. Finally we wrap up with conclusions and future work.

7.2 Background

Routing in the Internet

The Internet is a network of independent networks. These networks are referred to as Autonomous Systems (AS) and are administered by separate organizations. The routing inside an AS is managed by an Interior Gateway Protocol (IGP). Typically, IGP is a link state routing protocol like Intermediate System Intermediate System (IS-IS) or Open Shortest Path First (OSPF). In link state routing the network is modeled as a graph where nodes represent routers and arcs represent links connecting the routers. Each node collects information about network topology and calculates the shortest path to each destination node in the network.

In order to connect AS:es and exchange connectivity information, an External Gateway Protocol is used. The protocol currently in use is called Border Gateway Protocol version 4 (BGP4) [5]. BGP is a path vector protocol where an AS announces to its neighboring AS:es which networks it has a route to. In order to avoid routing loops the path of AS:es is included in the routing messages. In addition, the routing decision is also based on polices reflecting the relation the AS has with other AS:es, e.g. peering, customer or provider relations. When an AS has more than one route to a prefix, BGP has to select one route from the set of available routes. This is performed according to a decision process. The first step is to determine if there is a route to the egress point of the AS. Next BGP examines a number of BGP specific attributes.

If BGP still is unable to select one route, the shortest distance according to IGP is considered. This is sometimes referred to as hot-potato routing [10]. The final step is to use a vendor-specific tie-breaking. Figure 7.1 illustrates a simple example of a situation where a prefix is announced by two routers. In the example router R3 selects the route announced by R2 since it has the shortest IGP distance to R3. However, if the route announced by R2 is withdrawn the traffic towards network 192.168.0.0/16 injected in the network by R3 is shifted from the route announced

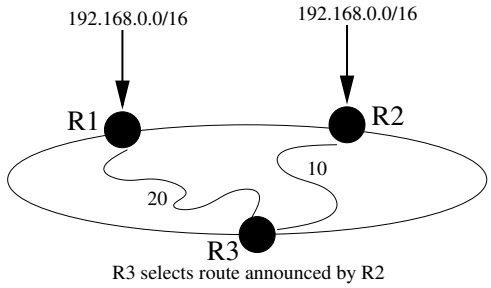

Figure 7.1: Routing scenario where the prefix 192.168.0.0/16 is announce by two peering points in the network. Router R3 has to select a route using the BGP decision process.

by R2 to the route announced by R1, causing a potentially massive change of the load on the links in the network.

7.3 Robust Routing Under BGP Reroutes

Robust Routing Under Uncertain Traffic Demands

Robustness, referring to the ability to cope with variations from the nominal operating conditions, is a key property of any engineering system. In this spirit, a robust network should be able to sustain acceptable performance despite foreseeable traffic variations and component failures. A common optimization objective in robust networking is to minimize the worst-case link loads, where worst-case should be understood as over all potential load variations or component failures. Our focus is on demand variations due to BGP reroutes.

Several methods for robust routing have been proposed recently [1, 2, 6, 8]. We will base our developments on the approach by Ben-Ameur and Kerivin [2] as we find it the most transparent. The method starts out from a standard arc-path formulation of multi commodity network flows

$$\begin{array}{ll} \text{minimize} & u_{\max} \\ \text{subject to} & \displaystyle\sum_{k}\sum_{\pi\in\Pi_k} r_{l\pi}\alpha_{\pi k}s_k \le c_l u_{\max}\ \forall l \\ & \displaystyle\sum_{\pi\in\Pi_k}\alpha_{\pi k} = 1, \qquad \alpha_{\pi k} \ge 0 \end{array} \tag{7.1}$$

Here, s_k is the aggregate traffic between source-destination pair k, Π_k is the set of all paths between source-destination pair k and $r_{l\pi}$ is an indicator variable taking the value one if path π traverses link l and zero otherwise. The optimization variables $\alpha_{\pi k}$ determine what fraction of the traffic between source destination pair k

that is routed across path π. The first set of constraints state that the total traffic across each link l is bounded by the link capacity times the maximal link utilization, while the second constraint states that all traffic must be routed across some path. The classical way of solving (7.1) is by column generation. Rather than explicitly enumerating all paths in the network, one starts out with a small subset of paths (*e.g.*, the shortest-hop routing) and then sequentially adds new paths to the problem to improve the optimization objective, see *e.g.*, [7] for details.

The robust multi commodity network flow problem is to find the routing that guarantees the smallest link utilization for all feasible traffic scenarios. We can formulate the problem as

$$\begin{array}{ll} \text{minimize} & u_{\max} \\ \text{subject to} & \displaystyle\sum_k \sum_{\pi \in \Pi_k} r_{l\pi} \alpha_{\pi k} s_k \leq c_l u_{\max} \, \forall l, \forall s \in \mathcal{S} \\ & \displaystyle\sum_{\pi \in \Pi_k} \alpha_{\pi k} = 1, \qquad \alpha_{\pi k} \geq 0 \end{array} \tag{7.2}$$

Depending on the nature of the traffic uncertainty set $\mathcal{S}$, this problem may or may not admit an efficient solution. If the traffic uncertainty is polyhedral $\mathcal{S} = \text{co}\{s^{(1)}, \cdots, s^{(V)}\}$, then (7.2) can be equivalently expressed as

$$\begin{array}{ll} \text{minimize} & u_{\max} \\ \text{subject to} & \displaystyle\sum_k \sum_{\pi \in \Pi_k} r_{l\pi} \alpha_{\pi k} s_k^{(v)} \leq c_l u_{\max} \, \forall l, v \\ & \displaystyle\sum_{\pi \in \Pi_k} \alpha_{\pi k} = 1, \qquad \alpha_{\pi k} \geq 0 \end{array} \tag{7.3}$$

There are at least two problems with this formulation. First, the traffic uncertainty sets are typically not given in vertex form, but as the set of solutions to a system of linear inequalities (cf. the demand uncertainty set $\mathcal{S}$ in Johansson and Gunnar [6]). Secondly, the uncertainty set may have many vertices, so that explicit enumeration is computationally unattractive. These two issues can be addressed similarly to the way column generation is used to avoid explicit enumeration of all paths in the nominal formulation: one starts out with a single traffic scenario in the uncertainty set, solves the routing problem, and then verifies whether the computed routing satisfies the link constraints for all feasible traffic loads. If this is not the case, one adds the traffic matrix that violates the constraints the most to the vertex description of the uncertainty set and repeats. The resulting method is a combined column- and constraint generation scheme, and is readily shown to have finite convergence (*e.g.* [2]).

A Model for Traffic Uncertainty due to BGP Reroutes

To describe traffic uncertainty under BGP reroutes, it will be convenient to be explicit about the source and destination node for each demand. Thus, rather than

using the notation s_k for the traffic between source-destination pair k, we will write s_{oe} to emphasize that the traffic originates at nodes o and is destined for egress point e. Let $E(p)$ be the set of egress points for prefix p (i.e., the set of peering points that could potentially announce prefix p) and, conversely, let $P(e)$ be the set of prefixes that can be announced by peers connected to egress node e. The total demand from node s exiting the system at node e can then be described as

$$s_{oe} = \sum_{p \in P(e)} d_{op} \delta_{pe}$$

with

$$\sum_{e \in E(p)} \delta_{pe} = 1, \quad \delta_{pe} \geq 0 \text{ and } \delta_{pe} = 0 \text{ for } e \notin E(p)$$

In this formulation δ_{pe} can be interpreted as the relative amount of traffic demand for prefix p that can be served via egress point e. At first, this model might seem counter-intuitive as the peering autonomous systems can only decide whether or not to announce a certain prefix and not influence the relative amount of demand for a specific prefix that it will allow to transit. However, as we will see shortly, the model serves its purpose. Now, assume that the internal routing is fixed. The utilization of link l can then be written as

$$u_l = c_l^{-1} \sum_o \sum_e \alpha_{loe} s_{oe}$$

where α_{loe} is the fraction of the traffic between nodes o and e that traverses link l. In terms of the notation in the previous section, if (o, e) is source-destination pair k, then

$$\alpha_{loe} = \sum_{\pi \in \Pi_k} r_{l\pi} \alpha_{\pi k}$$

Combining this with the expression above, we find

$$u_l = c_l^{-1} \sum_o \sum_e \alpha_{loe} \sum_{p \in P(e)} d_{op} \delta_{pe} \tag{7.4}$$

The worst-case traffic scenario is when prefixes are announced at peering points in a way that maximizes the maximum link utilization. From the expression above, we see that the worst-case situation is when prefix p is only announced at the egress e with largest value of $\alpha_{loe} d_{op}$ (i.e. when $\delta_{pe} = 1$ for this egress and zero for the others). Thus, in worst-case traffic scenarios generated by adjusting the prefix distributions to maximize the worst-case link utilization will be such that each prefix is announced by a single peer only, and thus compatible with realistic (and admissible) BGP configurations.

Optimizing Routing for BGP Reroute Uncertainty

We are now ready to summarize our procedure for finding a routing that is robust to BGP re-routes.

1. Generate a nominal traffic scenario set $\mathcal{S}$ by picking a single peering point for each prefix and computing the associated traffic matrix.
2. Compute the robust routing for the traffic scenario $\mathcal{S}$ by solving (7.3).
3. Fix the current routing and determine the prefix distribution that maximizes the utilization of the most loaded link by solving (7.4) for each link l. If the worst-case utilization is higher than predicted when optimizing the routing, add the corresponding traffic matrix to the scenario set $\mathcal{S}$ and return to step 2), otherwise terminate the algorithm.

Since the complete scenario set is finite, the algorithm has finite convergence. However, our computational experience, reported next, indicates that only a handful of iterations need to be carried out before the worst-case traffic scenarios are found and the optimal routing can be determined.

7.4 Analysis on Data From an Operational IP Network

In this section we evaluate our approach using traffic data from an IP network operator. We start with describing the network and highlight some properties of the routing and traffic data.

Data Collection and Evaluation Data Set

For the evaluation we have access to traffic data obtained from Netflow measurements as well as BGP routing information base and network topology. The data set was obtained from the Geant network [4] connecting European national research and university networks and consists of 23 nodes and 74 links. The measurements were conducted during a four month period and consist 15 minute flow export of sampled Netflow measurements with sampling rate of 1/1000; *i. e.* one packet of one thousand is sampled. In addition, a dump of the BGP routing information base from each day of the measurement period was conducted. The analysis in this paper was performed on data from one 15 minute measurement. More details about the network and traffic data can be found in Uhlig *et al.* [11].

Evaluation

Preliminary Data Analysis

Figure 7.2 shows the cumulative distribution of traffic in the Geant network classified by prefix. The prefixes are ranked by the amount of traffic sent towards

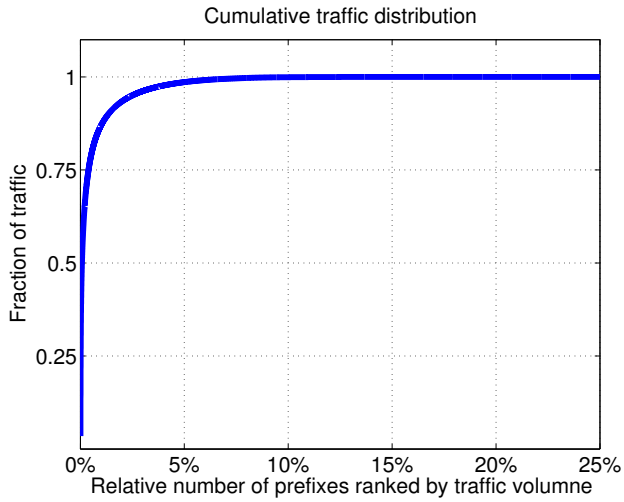

Figure 7.2: Cumulative traffic distribution

them during the measurement period. The figure reveals that only around seven percent of the prefixes have traffic routed towards them. The distribution of exit points for the prefixes is shown in the histogram in Figure 7.3. One can see that more than 60% of the prefixes are announced by five different locations and only three percent are announced by a single location. This could lead to a disruptive behavior in the traffic distribution since BGP might select another egress router for the traffic, with a potentially large impact on the load on internal network links. Figure 7.4 reveals that while most of the traffic is routed towards networks with only one exit point announced, 40% of the total traffic has multiple exit points and can thus be shifted around due to BGP reroutes.

Reducing the Number of Variables

Solving the optimization problem in Equation (7.4) for every prefix in the network would create a huge optimization problem since a typical backbone router has in the order of 160000 prefixes in its routing table. However, from Figure 7.2 we learn that only a small fraction of the prefixes account for the traffic in the network. Hence, by filtering out the prefixes with negligible traffic we are able to reduce the number of variables substantially. In our experiments we selected the prefixes that account for 90% of the traffic in the Geant network. Thus reducing the number of prefixes in our equations to 3600. In addition, since we consider the worst-case link utilization as optimization metric, we can treat links one-by-one, reducing the number of variables even further.

With these tricks we are able to reduce the number of variables to the order

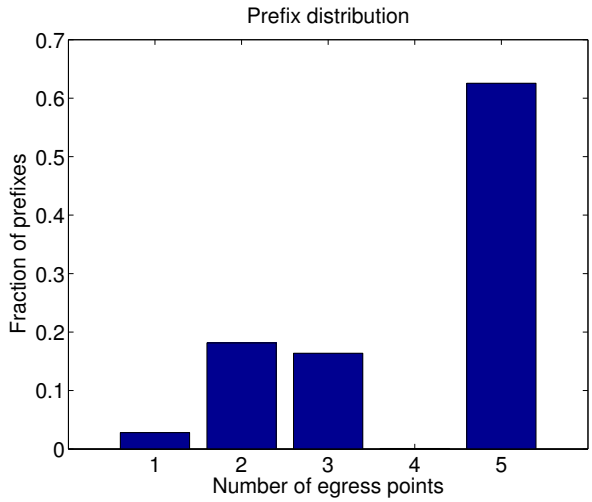

Figure 7.3: Number of prefixes with multiple exit points in the network

60000. Although this still constitutes a large optimization problem, most of the variables are uniquely determined by the constraints and the problem is readily solved on a regular desktop computer.

Experimental Results

The nominal traffic situation in our experiments is the traffic demands where all possible routes are announced in the network and link weights are set to the original values. In our experiments we have calculated the link loads for the following routing principles:

- **ROBUST:** the approach described in Section 7.3 where the worst case traffic scenarios from repeated optimization of Eqn.(7.4) are used to form the polyhedral $\mathcal{S}$.
- **MCNF_NL:** Multi commodity network flow routing using node-link formulation to minimize the maximum link utilization under nominal traffic.
- **MCNF_LP:** Multi commodity flow using a link-path formulation, i.e. solving problem (7.1), under nominal traffic.
- **SPF:** Shortest path first routing using the original link weights from the Geant network.

Figure 7.5 shows the utilization for the links in the Geant network under ROBUST, MCNF_NL and SPF routing for the nominal traffic scenario (in which the

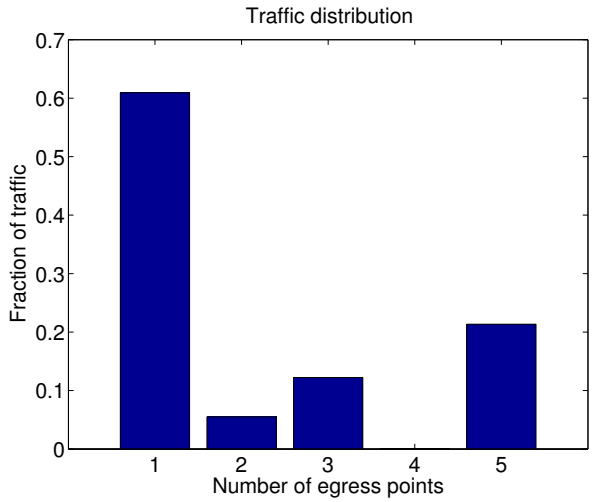

Figure 7.4: Number of bytes destined to prefixes with multiple exit points

robust routing coincides with MCNF_LP). We can see that although the node-link and link-path formulations achieve the same maximum link utilization, the robust routing achieves a better balance in the overall link utilization. This is due to that new paths are calculated using the dual variables of the link constraints in (7.3), which discourages routing across highly loaded links.

In Figure 7.6 we have plotted maximum link utilization under feasible traffic shifts for three routing configurations (SPF, MCNF_LP and ROBUST) and four scenarios (nominal traffic and three worst-case scenarios generated during the robust optimization). The robust routing is able to route efficiently in all three scenarios whereas the multi-commodity network flow routing optimized for the nominal traffic scenario suffers a substantial performance losses under BGP-reroutes, and performs on par with the original shortest-path routing.

Table 7.1 summarizes performance for each iteration of the algorithm in section 7.3. After four iterations the algorithm terminates with 758 paths. The algorithm has added 252 paths to be set up by MPLS in addition to the 506 shortest paths from link state routing.

7.5 Conclusions and Future Work

In this paper we have introduced a novel method to find critical traffic scenarios that can be used to find a routing setting that can route efficiently under all realistic traffic scenarios that can occur in a network due to inter domain rerouting. The scenarios are identified by finding the worst case setting of the Inter-domain rout-

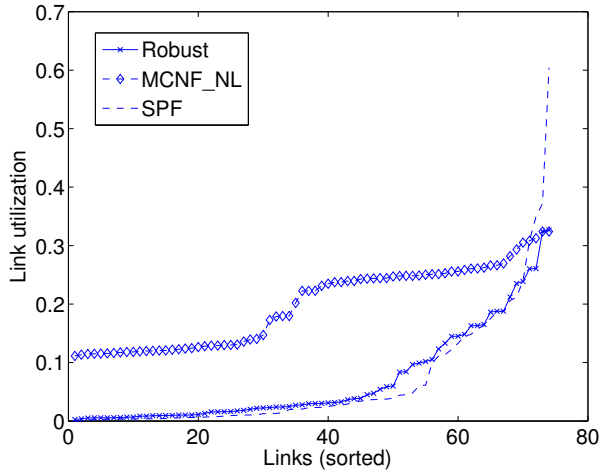

Figure 7.5: Link utilizations in the Geant network for robust, optimal and shortest path routing using the real link weights in the nominal traffic scenario.

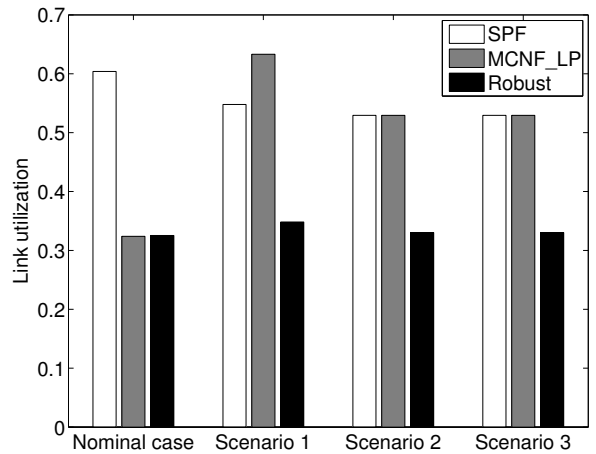

Figure 7.6: Maximum link load for SPF, MCNF_LP and ROBUST evaluated for the critical traffic scenarios generated by the robust routing algorithm. The maximum link utilization for the non-robust routings is around 0.6 (in the Nominal case for SPF, and in Scenario 1 for MCNF_LP) while it never exceeds 0.33 for the robust routing.

Iteration	1	2	3	4
$u_{\max}$	0.6	0.39	0.37	0.33
Paths	506	705	730	758

Table 7.1: Maximum link utilization and number of paths in network for each iteration of the algorithm

ing by solving a convex optimization problem. We show that the robust routing is able to minimize link load under a number of plausible traffic scenarios.

Our approach only considers changes in the external routing. Many occurrences of massive traffic shifts in a network stems from changes in the internal topology. To devise an algorithm that take these changes into account is a much more challenging problem and is one avenue of future work. Further, our results has only been tested on one sample of traffic and routing data from one network. A more interesting scenario is to test our algorithms on a time series of data and for data from other networks. For instance Figure 7.4 reveals that only 20 percent of the traffic is routed to prefixes announced in five places. A network with a larger fraction of traffic routed to prefixes announced in multiple places would have illustrated the benefit of our approach clearer. Another property of the Geant network that caused some problems in our experiments was that the links in the network have highly diverse capacity, indicating that it could be relevant to study other performance measures than worst-case link utilization.

Acknowledgment

This work was supported by SICS Center for Networked Systems, the Linnaeus center ACCESS and the Swedish Research Council.

Bibliography

[1] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *Proc. ACM SIGCOMM*, pages 313–324, Karlsruhe, Germany, August 2003.

[2] W. Ben-Ameur and H. Kerivin. Routing of uncertain demands. *Optimization and Engineering*, 6(3):283–313, 2005.

[3] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, 2002.

[4] *Geant*. http://www.geant.net.

[5] S. Halabi and D. McPherson. *Internet Routing Archtectures*. Cisco Press, 2001.

[6] Mikael Johansson and Anders Gunnar. Data-driven traffic engineering: techniques, experiences and challenges. In *Proc. Broadnets 2006*, San Jose, California, October 2006.

[7] M. Pioro and D. Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, 2004.

[8] A. Sridharan, R. Guérin, C. Diot, and S. Bhattacharyya. The impact of traffic granuarity of robustness of traffic aware routing. Technical report, University of Pennsylvania, 2004. Available via http://einstein.seas.upenn.edu/mnlab.

[9] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan. Traffic matrix reloaded: The impact of routing changes. In *Proc. Passive Active Measurements*, Boston, Massachusetts, USA, 2005.

[10] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. In *Proc. ACM SIGMETRICS*, pages 307–319, New York, USA, June 2004.

[11] S. Uhlig, B. Quoitin, S. Balon, and J. Lepropre. Providing public intradomain traffic matrices to the research community. *ACM SIGCOMM Computer Communication Review*, 36(1), January 2006.

[12] Hao Wang, Haiyong Xie, Lili Qiu, Yang Richard Yang, Yin Zhang, and Albert Greenberg. Cope: traffic engineering in dynamic networks. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 99–110, New York, NY, USA, 2006. ACM Press.