Robust Routing under Dynamic Traffic Demands

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

> Submitted by Himanshu Agrawal M. Tech.

Electrical and Computer Engineering College of Science, Engineering and Health RMIT University August 2009

Statement of authorship

The candidate declares that,

- except where due acknowledgement has been made, the work is that of mine alone.
- the work has not been submitted previously, in whole or in part, to qualify for any other academic award.
- the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.
- any editorial work, paid or unpaid, carried out by a third party has been acknowledged.
- Ethics, procedures and guidelines have been followed.

Signed by the Candidate Himanshu Agrawal

Date

Acknowledgement

First and foremost, I am grateful to Professor Andrew Jennings, my senior supervisor for guiding me through out my studies. Professor Jennings is always there to support, to guide, and to help, with his calming influence and amazing wisdom. Without his continued support, this work would not have been a reality. I have learned so much from you, and thank you Prof. Jennings.

I am also thankful to my second supervisor Dr. Mark Gregory for useful insights given from time to time during this study. I would like to mention his guidance thankfully, during initial stage of my research when he was my senior supervisor. I am sincerely thankful to Pietro Belloti, who gave me invaluable suggestions through communication.

I am also indebted to my parents for their encouragement and their continued support throughout my research program. I specially thank my wife Nidhi and two wonderful kids Yashi/Aditi for their unconditional love through out my studies. I thank my research colleagues and friends who helped me directly or indirectly during my studies.

Finally, I am very thankful to the School of Electrical and Computer Engineering of RMIT University for the financial support and scholarship which was highly instrumental for my dreams and aspirations of pursuing a research degree.Also, I would like to thank research administration office in my school. Thanks everyone for the kind support during my research carrier.

Abstract

In order to provide service reliability with reasonable quality, it is essential for the network operator to manage the traffic flows in the core network. Managing traffic in the network is performed as routing function. In the traditional traffic management, network operator can tune routing parameters to simply manage the traffic. But traditional routing methods are not designed to handle the sudden fluctuations in the traffic. As a result, this may apparently lead to the traffic congestions in some parts of the core network, leaving other part underutilized. In this thesis we explore issues related to the routing robustness in the face of traffic demand variations.

We investigate different routing methods for efficient routing using maximum link utilization (MLU) as a performance metric. The primary advantage of using link utilization is its ease to compute the network performance on real network data and synthetic data. Overloaded links might result in Quality of Service degradation (e.g. larger packet delay, packet losses etc.), so MLU might be a useful measure of network performance. For the experimentation, we have used unique data from the real operational network available in the public domain and the random data for large network topology instances.

Furthermore, we propose a simple routing algorithm called Robust Routing Technique (RRT) to implement a robust routing mechanism. This mechanism allows network operator to satisfy the networking goals such as load balancing, routing robustness to the range of traffic demand matrices, link failures or to the traffic changes caused by uncertain traffic demands. Simulation experiments with real network topologies and random topologies demonstrate that our routing solution is simple (for routing) and flexible (for forwarding). K-Shortest path implementation in RRT can be extended for Multi Protocol Label Switching.

Finally, we evaluate the performance of robust routing under dynamic traffic demands. We formulate the problem as a multi commodity flow problem using linear programming. We use congestion ratio to define the robust routing performance. We provide a variant to the existing robust routing mechanisms by modelling traffic demand due to Distributed Denial of service attacks or worms. Simulation results are compared with the popular OSPF traffic engineering algorithm to provide effectiveness to the proposed routing scheme. Simulation results are compared with the popular OSPF traffic engineering algorithm to provide effectiveness to the proposed routing scheme.

Contents

	Statement of Authorship		
	Acknowledgements		
	Abstract		
	List of Figures		
	List of Tables		
	List of Publications		
1	Introduction		
	1.1	Basics of Internet routing	2
	1.2	Traffic Engineering in IP Networks	4
	1.3	Motivation in Developing Robust TE solution	6
	1.4	Contribution of the dissertation	10
	1.5	Outline of the dissertation	11

2	Bac	ground and Literature Review	13
	2.1	Overview	13
		2.1.1 Traffic demand and matrix	14
		2.1.2 IP based TE	15
		2.1.3 MPLS based TE	17
		2.1.4 Intradomain vs. Interdomain TE	19
		2.1.5 Offline vs. Online TE	21
	2.2	2 Advances in TE: related work	
		2.2.1 Proactive TE	23
		2.2.2 Reactive TE	25
	2.3	Discussion and Summary	26
3 Robust Routing Problem		ust Routing Problem	29
	3.1	Overview	29
	3.2	Maximal flow routing problem	30
	3.3	Shortest path routing problem	31
	3.4	Robust shortest path problem	32
	3.5	Multi commodity flow problem	33
4	Roi	ting Problem with Robustness to Traffic Demand	35
	4.1	Motivation	35
	4.2	Notations and symbols	37
	4.3	Oblivious routing	40
	4.4 Robust routing design		41
		4.4.1 Mathematical formulation	41
		4.4.2 Problem statement	41
		4.4.3 Network Model	43

		4.4.4	Algorithm	44
	4.5	Perform	mance evaluation	46
		4.5.1	Data	46
		4.5.2	Topologies	46
		4.5.3	Methods of traffic matrices (TMs) generation	46
		4.5.4	Simulation set up	47
	4.6	Result	and analysis \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots	48
		4.6.1	Robust routing environment on set of traffic matrices .	48
		4.6.2	Routing performance on larger random topologies	51
	4.7	Discus	sion	54
	4.8	Summ	ary	56
5	Rob	oust Ro	outing Under Dynamic Traffic Demand	58
	5.1	Introd	uction	58
	5.2	Perform	mance metric under demand uncertainty	60
	5.3	Optim	al routing under dynamic traffic demand	61
	5.4	Traffic	model	63
		5.4.1	Hose model	63
		5.4.2	Polyhedral model	63
		5.4.3	Bertimas and Sim. model	64
	5.5	Robust	t routing in the face of voluminous traffic demand \ldots	65
		5.5.1	Robust algorithm	65
		5.5.2	Mathematical formulation	65
	5.6	Perform	mance evaluation	68
		5.6.1	Experimental setup	68
		5.6.2	Demand model under uncertainty due to DDoS and worms	69

		5.6.3 Results \ldots	70
	5.7	Discussion	74
	5.8	Summary	75
6	Con	nclusion and Future Directions	77
	6.1	Conclusion	77
	6.2	Future directions	79
	App	pendix	81
	Bibliography		

List of Figures

1.1	Traffic trace of worms 24 hours	8
1.2	Traffic trace of worms 1 week	9
1.3	Traffic trace of DDoS attacks 24 hours	9
1.4	Traffic trace of DDoS attacks 1 week	10
2.1	Broad Segmentation of Traffic Engineering	14
2.2	Traffic engineering with ECMP rule	17
3.1	Multi commodity flow problem	34
4.1	Schematic diagram for the robust routing model	44
4.2	Performance evaluation on real network topologies. The value on the y axis denotes MLU, which is expressed as a fraction instead of percentage. Link weights are assigned as per inverse capacity rule and the traffic margin is set to 1.1	49
4.3	Performance evaluation on real network topologies. The value on the y axis denotes MLU, which is expressed as a fraction instead of percentage. Link weights are assigned as inverse ca-	
	pacity and traffic margin is set to 5.0	50

4.4	Performance evaluation on real network topologies. The value on the y axis denotes MLU, which is expressed as a fraction instead of percentage. Link weights are assigned as per inverse capacity rule and traffic margin is set to 10.0	50
4.5	Performance evaluation on real network topologies. The value on the y axis denotes MLU, which is expressed as a fraction instead of percentage. Link weights are assigned randomly and traffic margin is set to 5.0	51
4.6	Maximum Link Utilization of RRT algorithms on GT-ITM us- ing pure random graphs	52
4.7	Link Utilization on GT-ITM random graph r10 (number of nodes $=10$)	53
4.8	Link Utilization on GT-ITM random graph r30 (number of nodes =30)	53
4.9	Link Utilization on GT-ITM random graph r50 (number of nodes $=50$)	53
5.1	Congestion ratio for ISP topology AS1221	71
5.2	Congestion ratio for ISP topology AS6461	71
5.3	Congestion ratio for ISP topology nsf	71
5.4	Congestion ratio for ISP topology AS4755	73

List of Tables

2.1	Summary of related research work	28
4.1	Notations used in the problem formulation	41
4.2	RRT Algorithm	45
4.3	Maximum Link Utilization of RRT algorithms on GT-ITM us- ing pure random graphs	52
4.4		54
5.1	Notations used in the problem formulation	61
5.2	Routing under dynamic demand	65
5.3	congestion ratio for Robust and Optimal routing solutions $\ . \ .$	72

List of Publications

- H. Agrawal, A. Jennings, Robust Solution for Optimal Routing, Proceedings Advanced Networks and Telecommunication Systems conference, ANTS 2007.
- H. Agrawal, A. Jennings, M. Gregory, D. Channa, Robust Routing, Proceedings Seventh IEEE/ACIS International Conference on Computer and Information Science, 2008.
- H. Agrawal, A. Jennings and M. Gregory, Robust Traffic Engineering, Proceedings IEEE Advanced Networks and Telecommunication Systems conference IEEE-ANTS 2008.
- H. Agrawal, A. Jennings, Q. D. Bui., A hybrid approach for Robust Traffic Engineering, Proceedings 4th International Symposium on Wireless Pervasive Computing, ISWPC 2009

Chapter 1

Introduction

The Internet has evolved from an ARPNET in 1969. Since it's inception as a simple network when it was used for file transfer and email services by the research community, Internet has grown with a phenomenal rate and now has been adopted by each segment of society. Millions of hosts are communicating using Internet for user applications such as voice over IP, video on demand, multimedia and real time applications, IPTV etc. Today, Internet is a multi service network that can support many types of applications with potentially high demand in bandwidth. Rapid development in the communication hardware is adding resources e.g. high speed switches, routers and high-speed optical links to the Internet thus making services relatively cheaper. Particularly, the advent of optical fibre technology has offered Internet Service Providers (ISPs) an opportunity of over-provisioning the bandwidth in the network. Nevertheless, this approach is currently limited to the core network only and the growing traffic demand over the global network still cannot be managed efficiently using existing tools.

In this thesis, we study the methods to improve the routing in Internet. We investigate different routing methods based on traffic situations and evaluate the network performance on real and synthetic data. We further develop a routing model that can be used to understand variations in the traffic demand due to routing shifts within the ISP network, routing changes occurred outside the ISP network or sudden demand variations.

1.1 Basics of Internet routing

Internet routing is an important network layer activity that guides the packets through the communication subnet to their correct destination [18]. Routing decisions are mainly based on the datagram or virtual circuit. In a datagram network, router uses routing algorithm to route the packets. Two packets from the same node may follow different routes. A routing decision is necessary for each individual packet. In a virtual circuit network, routing algorithm works to select a path between two nodes for the virtual circuit. All packets of the virtual circuit subsequently use this path.

More precisely routing in a data network involves set of complex routing algorithms to provide services. The complexity of the routing algorithm arises due to variety of issues. First, routing is not simply forwarding packets, rather attempting smooth coordination between all the nodes in a subnet. Second, routing system must be able to redirect or reroute the traffic in case of link and node failures. Third, routing algorithm must provide alternative routes when some areas in the network become congested. Routing algorithms may be classified as *static* or *dynamic*, based on whether they change routes in response to the input traffic patterns. In static routing, paths between the origin-destination pairs remain fixed regardless of traffic conditions in the network. It may change only in response to link or node failure. Hence *static* routing algorithms could not provide adequate performance in terms of throughput under varying traffic input patterns. Therefore static routing algorithms are recommended for small networks. A large network on the other hand will use some form of *adaptive* routing, where the path to route traffic may change in response to congestion.

Proper selection of routes require a detailed analysis of traffic flows and available resources. At the same time, if each individual traffic source makes uncoordinated routing decisions based on the same network data, the result may be the sudden and simultaneous transferral of all traffic from one over-used link to another (underused) link in a way that may cause even more serious congestion. This can be compared with the rush hour traffic that responds to a radio traffic report by diverting traffic from the slightly congested freeway to the single-lane country road.

A better model, therefore, might have paths selected in coordination with a centralized traffic control station. This sort of model is applied very successfully in vehicular traffic engineering when a city or motorway network is controlled through an operational headquarters that can control speed limits, traffic lights, diversions and lane assignments. However this may lead to increased overhead of soliciting a path for an individual host.

On the other hand, significant traffic management can usefully be per-

formed within the core of the network where the traffic volumes are greater. Here individual flows from host to host are bundled together and treated in the same way for forwarding down to the routes that are not necessarily the shortest. The easiest way to handle this is through a process known as tunneling. This tunnel is a well-defined path from one point in the network to another. Traffic flows may be injected into the tunnel at one end to emerge at another end. ling. This tunnel is a well-defined path from one point in the network to another. Traffic flows may be injected into the tunnel at one end to emerge at another end.

1.2 Traffic Engineering in IP Networks

One of the techniques, adopted by Internet Service Provider (ISP) to manage the network resource, is Traffic Engineering (TE). TE is defined as large scale network engineering dealing with network performance evaluation and optimization [8, 10]. A more straightforward explanation is -"to put the traffic where the network bandwidth is available".

In recent years, the three major issues that have attracted the attention of research community in TE approaches are as follows:

1. Quality of Service (QoS): Emerging applications e.g. voice over IP, video on demand and multimedia and streaming applications not only have bandwidth requirement, but also need service guarantees in terms of end- to- end delay, jitter and packet loss probability between end users. These QoS requirements thus impose new challenge to the Internet Service Providers (ISPs) and thus need to be satisfied by designing a good TE tool.

- 2. Resilience: In the context of network engineering, resilience may be defined as the ability of the network to cope with variation in routing parameters. Given the fact that the network node or link failures are still a frequent event on the network, TE solutions have to consider how to minimize the impact of failure on network performance and resource utilization.
- 3. Security: Third, different security issues related to wired and wireless networks. We focus on the second issue i.e. resilience. We propose an approach for the routing robustness under dynamic traffic demand.

Existing TE solutions assume that traffic matrix (TM) is accurate and network is operating under normal conditions. However estimation of accurate TMs is far from trivial due to dynamic nature of Internet traffic. Moreover, network failures in particular often occur in the core network. As a result dynamic traffic demand and network failures may cause TE performance to be unpredictable and thus make the network management task more complicated. It is therefore necessary to make TE more robust in order to maintain the expected performance when any of those situations take place. In addition to achieve the expected performance, the other benefit of robust approach is that only one relatively stable configuration may be efficient without frequent changes in response to the occurrence of any unexpected situation.

In contrast to the traditional TE solutions, robust TE solutions are constrained by the variations in the traffic demand. We study the routing problem considering the variations in the traffic demand. Traffic demand refers to the amount of traffic between pair of nodes. Traffic demand is considered to have two components of behavior: one, a stable and predictable traffic component due to usual traffic caused by daily demand fluctuations and the other, an abrupt, uncertain behavior due to network equipment failure (e.g. node or link failure), malicious attacks (e.g. denial of service, worms, viruses), external routing changes (e.g.routing through BGP) or new spontaneous overlay services (e.g. P2P). This can be termed as demand uncertainty. In a robust version of TE, demand uncertainty is directly taken into account within routing optimization, computing a single routing configuration for all demands within an uncertainty set. The idea of robust TE is to model these network conditions as separate scenarios and apply an appropriate single routing configuration that may perform well under any network conditions.

1.3 Motivation in Developing Robust TE solution

In the past few years, there have been significant advances in the traffic engineering methods, from both academia and industry. Popular TE methods like OSPF weight optimizer (OSPF-TE) and MPLS multicommodity flow optimizer [1] have shown reduction in the maximum utilization over pure shortest path routing. Nonetheless, current TE methods in this area have following challenges to the researchers:

1. First, our work is largely motivated by previous studies on the routing algorithms under changing traffic demand [5,11,12,13,40]. There are two major classifications: proactive and reactive TE. The proactive TE algorithm optimizes the routing performance on collected traffic samples. Proactive algorithms perform efficiently when traffic is stable but can not readjust to handle unpredictable traffic. On the contrary, reactive TE algorithms adapt to the sudden and abrupt traffic demand quickly. Reactive solutions are responsive, but the major weakness is the stability of resulting routing configuration. We are motivated to combine the best of the two worlds:both proactive and reactive routing solutions in the robust routing framework.

2. Second, we are motivated by the fact that sudden and abrupt traffic variations may be contributed by anomalous traffic due to Distributed Denial of Service attacks or worms. Previous studies on robust TE solutions are missing the effect of DDoS and worms on robustness of solution.

A DDoS attack in it's simplest form can traget any IP address and if the attack is strong enough, it is likely to be successful. This may potentially affect not only the individual user machines but large and small business and ISPs and governments offices also that rely on networking by disrupting normal network traffic. As a result this may lead to a more serious problem of network traffic congestion if the attacks are distributed across large network .Even if your computer is protected by NAT, it is vulnerable to DoS attacks.

History of network based denial of service attacks [50] show that the problem first became evident in October 1986 when the Internet suffered a series of congestion collapses [54] and addressed subsequently



Figure 1.1: Traffic trace of worms 24 hours

by the design of TCP congestion control mechanism [29]. End hosts were sending more traffic than could be supported by interconnection network.Since we are emphasizing the variation of traffic caused by DDoS attcks hence the detail discussion on DDoS is ommitted.It can be seen in the following Figures 1.1 to 1.4, the proportion of traffic due to DDoS and worms shown under UNKNOWN_UDP. This data is taken from [20].CAIDA used a technique called backscatter to measure the DDoS attacks.

The four figures show the traffic variations on different time scales and indicates the proportion of traffic variation due to DDoS attacks and worms.Figures 1.1 to 1.4 show the traffic pattern during 24 hours and a week for a large backbone network [20].At this level of traffic aggregation,







Figure 1.3: Traffic trace of DDoS attacks 24 hours



Figure 1.4: Traffic trace of DDoS attacks 1 week

the traffic fluctuations may be predictable.

We define robustness in traffic engineering in two ways. First, we use an uncertainty parameter to test routing algorithm on a range of traffic demand matrices. Second, we introduce demand uncertainty to define abrupt traffic behavior e.g. distributed denial of service(DDoS) attacks, worms and virus propagation. Applying a single routing configuration to the normal traffic demand condition as well as dynamic traffic demand gives a novel approach.

1.4 Contribution of the dissertation

Having outlined the problem to be addressed in the thesis, we now summarize the main contribution of the thesis.

- * We study the problem of robust routing so as to evaluate the Maximum Link Utilization(MLU). We formulate the robust routing problem under dynamic traffic demand conditions. We model traffic demand to capture the variability of internet traffic. In our preliminary experiment, we compare the OSPF Optimizer with our proposed robust routing technique (RRT). We consider real data network topologies provided by Rocketfuel [65]. Simulation results show that proposed robust routing technique (RRT) performs efficiently on a range of traffic matrices. In addition K-shortest path implementation in RRT may be further extended for Multi Protocol Label Switching(MPLS).
- * We solve routing robustness as an optimization problem. We introduce the new idea of considering the effect of DDoS and worms as part of the robust routing problem. In this part we perform several experiments on ns2 to generate a file which forms an input to our AMPL script that runs the mathematical formulation of problem. We model dynamic traffic demand e.g. due to DDoS or worms using polytopes to represent uncertain traffic and a demand polytope to represent stable traffic. Simulation results are indicative of the performance improvement of RRT under traffic variations.

1.5 Outline of the dissertation

In the introduction, we have provided a brief description and motivation in the area of robust TE. In chapter 2, we provide a brief overview of traffic engineering, definition of the related terms and the related research work. In chapter 3, we define the multi commodity flow problem and the robust shortest path problem using Linear Programming to gain proper understanding. In chapter 4, we compare our proposed RRT with OSPT-TE as our initial experiment in this chapter. This provides a good starting point to understand robust routing under changing traffic demand. To further improve, we investigate and optimize robust TE in chapter 5.We use congestion ratio as a performance measuring criterion and compare the routing performance with oblivious routing. Chapter 5 also presents a theoretical framework for robust TE in the presence of voluminous traffic variations e.g. DDoS and worms. Finally chapter 6 concludes this thesis and gives some future research directions.



Background and Literature Review

2.1 Overview

In first part of this chapter, we give some basics of traffic engineering. This includes the definitions of traffic demand and traffic matrix. It also provides a brief overview of IP routing, intradomain TE, interdomain TE and MPLS based TE.In the second part, we have surveyed the recent research work in TE. Internet Traffic Engineering process can be segmented based on the aspect of optimization, routing and time scale of operation.Firstly, in terms of optimization, traffic engineering can be termed as Intradomain and Interdomain traffic engineering.Secondly, based on routing, we have IP and MPLS based TE.Finally, based on the time scale of operation, TE can be sub classified as offline and online or proactive and reactive TE. Figure 2.1 shows the segmentation of TE.



Figure 2.1: Broad Segmentation of Traffic Engineering

2.1.1 Traffic demand and matrix

IP traffic could be represented in two ways.One way is to represent the traffic inferred from measurement as an aggregate traffic between possible source and destinations to the network address or autonomous system level.Traffic Matrix(TM reflects the amount of traffic data that flows between all possible pairs of origin and destination in a network [49].Such end to end traffic representation would result in an extremely large traffic matrix [28] and makes it difficult to populate such a model.

Alternatively, IP traffic could be aggregated to point to point demands between edge links or routers in the ISP backbone, suggested in the context of MPLS network in [73].However this approach has a fundamental difficulty in dealing with traffic that traverses multiple domains.

As a result IP traffic demands are naturally modelled as point to multi-

point volumes. Traffic demand is defined as a logical tuple: ingress router, egress router and the flow between these two routers. In a communication network, the traffic that transits through the network has an origin where that particular traffic flow enters the network and a destination where it exits the network. In the context of Internet, the traffic matrix is obtained as an estimation of data collected as link counts using SNMP. SNMP provides these data sets via incoming and outgoing byte counts computed per link every 5 minutes. The concept of TM [49,70] was originally associated with Intradomain TE, where the ingress/egress points are fixed. In this case the overall traffic demand on the network can be represented by a TM. For instance, each element, t(i, j) of TM being the total bandwidth demand of individual traffic flows from ingress node i to egress node j. If we have only a single measurement, it can be interpreted to be the traffic matrix. On the other hand, in the case when there is a time series of values available, the measurement can be interpreted as samples from a stochastic variable whose expected value is the traffic matrix. To solve the routing optimization or load balancing problem, the traffic matrix is supplied as an input to the routing algorithm.

Existing techniques based on traffic measurement provide views of the effect of traffic demand in terms of end to end performance by active measurement of delay. loss and throughput [28, 56, 68].

2.1.2 IP based TE

IP routing can be defined as the process of directing traffic flows between ingress and egress routers based on the combined action of Interior Gateway Protocol (IGP) and Border Gateway Protocol (BGP). The Internet is an interconnection of several Autonomous Systems (AS). Routing in an AS is performed by the interplay of interdomain and intradomain routing. The most popular Interior Gateway protocols are OSPF and IS-IS. With these link state protocols, each router learns the entire network topology and uses Dijkstra's algorithm [23] to compute shortest path to all other routers in the network. Routers on the border of an AS exchange reachability information uses BGP. Inside each AS interior gateway protocol (IGP) determines shortest paths between routers in the network. BGP determines the set of best egress points for a destination prefix. When there are multiple equally good egress points, as information to BGP, IGP decides and order these egress points based on the distance from ingress point to complete the routing process [38].

In the process of routing, the route selection is based on analysis of traffic flows and available resources. If each individual traffic source makes uncoordinated routing decisions based on the same network data, the result may be the sudden and simultaneous transferral of all traffic from one over-used link to another (underused) link in a way that may cause even more serious congestion.

The goal of TE in IP network is to minimize congestion [59] and make the better use of available network resources by adapting the routing to the current traffic situation. The two most commonly used protocols at the IP layer are Open Shortest Path First (OSPF) [51,52] and Intermediate System to Intermediate System (IS-IS). Routing decisions are based on the link cost and a shortest (least cost) path calculation. Equal-cost multi-



Figure 2.2: Traffic engineering with ECMP rule

path (ECMP) extension of OSPF protocol can also be used to evenly distribute the traffic over several paths when there are multiple least cost paths.We use the Figure 2.2 to explain the ECMP process.When the traffic flow arrives at an ingress router n, flows get evenly split and follow the equal cost path via the egress routers $n(1), n(2), \dots, n(k)$ to reach router t.

2.1.3 MPLS based TE

The concept of MPLS-TE was first introduced in [9, 10]. MPLS based TE can provide an efficient paradigm for traffic optimization with explicit routing and arbitrary splitting of traffic, which is highly flexible for both routing and forwarding purposes. However since traffic trunks are delivered through dedicated label switched paths (LSPs), hence scalability and robustness remain as open issues in MPLS-based TE.

* *Scalability*: total number of LSPs (assuming full mesh network) within a domain increases exponentially between ingress-egress router

pairs. This means the overhead of setting up explicit label switched path may be very high, particularly for a large-size network.

* Resilience: path protection mechanisms (e.g. using back up paths) are necessary in MPLS-based TE as otherwise traffic cannot be automatically delivered through alternative paths in case of any link or node failure in active LSPs.

MPLS-based TE vs. IP based: The first IP based TE was proposed by Fortz and Thorup [30,31]. They used link weight optimization. Given a network topology and traffic demand, the basic idea in the link weight optimization is to use set of link weights as information to interior gateway protocol (IGP) to control the Intradomain TE performance objective. On the contrary, MPLS- based TE provided fine grained path selection using explicit routing for individual flows, which cannot be achieved by IPbased TE, as the change in IGP weight may affect the routing pattern of entire set of traffic flows.

- 1. In comparison to the MPLS- based approach, IP-based TE lacks flexibility in the path selection, since explicit routing and uneven splitting are not supported.
- 2. However IP- based approach offers better scalability and availability resilience than MPLS- based TE because there is no overhead of setting up explicit LSPs and also because traffic can be automatically delivered via alternative shortest paths in case of node or link failure with explicit provisioning of back up paths. However given this type of auto-rerouting of traffic in the IP based environment, single link failure may introduce dramatic changes in the traffic distribution

(thus causing new traffic congestions) even across multiple domains.

3. In [67], R. Teixeira et al. pointed that with the combined IGP/BGP decision making in IP routing, an intradomain link failure may cause transit traffic to shift to alternative egress points due to hot potato routing effect. This attributes low TE robustness for IP based approach as compared to MPLS based approach, where a single link failure has minimal impact on other primary LSPs.

2.1.4 Intradomain vs. Interdomain TE

In this section, we discuss the segmentation of TE, based on the aspect of optimization [37]. Intradomain Traffic engineering refers to the optimization of customer traffic within single ISP domain. Recent development in IP based TE solutions has challenged the MPLS- based approaches in that Internet traffic can also be effectively tuned through native hop-byhop routing, without the associated complexity and cost of MPLS.

There are number of research publications in the Intradomain TE category. Fortz and Thorup [30, 31] claimed that by optimizing OSPF link weights for the load balancing, network service capability can be improved significantly in comparison to the conventional approach of setting link weight inverse of link capacity. Another link weight optimization approach was proposed by Wang et al. [75], without the necessity of ECMP splitting. Their approach is to divide the physical network into logical routing planes, each being associated with a dedicated link weight configuration. The basic strategy of this approach is to emulate MPLS unequal splitting of flows by partitioning the overall traffic demand at the edge of the network so that traffic within different partitions is delivered through dedicated routing planes.

M. Ericsson et al [26] proposed a Genetic Algorithm (GA) approach to solve Intradomain IP-based TE problem. M. Ericsson et al claimed the performance close to that in [30,31] by properly tuning GA parameters. In addition, Retvari et al. raised some practical issues in OSPF traffic engineering, such as exploiting knowledge of link capacity and reasonable range of OSPF link weight values [61] and formulated the TE as the minimum cost maximum throughput problem and resulting link weight routing configuration provides plausible basis to build a practical IPbased TE solutions.Edge based traffic engineering was proposed in [72]

Moreover, a near-optimal routing solution was proposed by A. Sridharan et al. [66]. They present a simple local search heuristic to realize a near optimal traffic distribution without changes to routing protocols and forwarding paradigm. Moreover, this research paper defines a routing mechanism for selecting a set of allowable next hops by carefully selecting this subset from the set of next hops corresponding to the shortest paths. A combinatorial algorithm was proposed for optimal routing in Internet Protocol networks using open shortest path first(OSPF),routing information protocol(RIP) and interior gateway protocol(IGP),in [60].

Another emerging research area is Interdomain TE that has evolved from its Intradomain counterpart. Interdomain TE uses Border Gateway Protocol (BGP) to exchange routing information between two Autonomous Systems (ASs). BGP performs this interdomain TE by routing information advertised by adjacent ASs. We note that the change in the routing configuration of one AS might affect the routing decisions of nearby ASs, and this can propagate in cascade. This often causes routing instabilities across the whole Internet, where a single change in interdomain path may take several minutes to converge [32]. Therefore, interdomain routing must ensure stable traffic distribution and fast routing convergence [32, 34]. Some recent research proposals on interdomain TE have provided some guidelines to achieve predictable traffic flow changes, limiting the effect of neighboring domain and minimizing the overhead of routing changes [27, 74].

2.1.5 Offline vs. Online TE

Another important TE segmentation is based on the timescale of traffic manipulation and availability of the traffic matrix. In some situations, Internet Service Providers(ISPs) can predict traffic before routing optimization is performed. ISP may use either traffic monitoring or measurement tool to forecast TM. Given the traffic matrix for the specific network, ISP can perform off line TE. One important issue in off line TE is the average duration between two consecutive TE cycles. Depending upon the customer Service Level Specification the off line TE cycle may be weekly or monthly. The major weakness of off line TE is the lack of adaptive traffic manipulation according to traffic and operational network dynamics such as traffic burst and network failures.

In some other cases, ISP might not be able to predict the overall TM in advance and this requires that the ISP perform on line TE that does not require any knowledge about future traffic demand. Online TE offers on a timescale of hours or even minutes to respond to dynamic traffic fluctuations. A practical concern for ISP is how to make sure such a dynamic routing system is converged without human intervention. Online TE should balance traffic load evenly in case of random incoming traffic demand in future. Rerouting may provide a solution to reserve the bandwidth for new and future traffic. As a result of rerouting, competing flows might interfere with each other and cause traffic instability and service disruption. Also, due to uncertainty in the traffic pattern on smaller timescale, on line TE may pose difficulties in handling future incoming traffic based on the current state of the network. Therefore, a promising approach is to consider both off line and on line TE together as complementary to each other.

2.2 Advances in TE: related work

As mentioned earlier, the aim of traffic engineering is to optimize the usage of network resources subject to the traffic constraints. However, the traffic situation may change in the network over time, e.g. due to changing user behavior, new applications or changes in the routing systems. To handle the traffic demand changes, there are basically two major segmentations: Proactive traffic engineering uses fixed routing settings to handle a wide variety of traffic situations. 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 stable but performance will not be optimal in all situations.

Reactive traffic engineering solutions, on the other hand continuously

monitor the state of network and adapt routing to handle changes in the traffic situation. This approach enables the network to handle unpredictable changes in traffic demand and the network to operate at an optimal point at all times. However, this requires the network operator to monitor the network state continuously which imposes extra overhead. As a result the routing solution may not be very cost-effective.

2.2.1 Proactive TE

The first category of algorithms is stable robust routing techniques or proactive TE algorithm. One of the earliest and benchmark research publications in this category was proposed by Applegate and Cohen [4, 5]. They proposed oblivious routing that may perform with limited or no knowledge of traffic data and define linear algorithms to optimize the worst-case performance for different sizes of traffic uncertainty sets, aiming to handle dynamic changes. Furthermore, they provide a lower bound on the performance for the routing under all possible traffic situations. In [31] Fortz and Thorup use a search heuristic for OSPF/IS-IS in finding suitable link weight settings to a given traffic situation. Azar et al [11] have shown that the routing performance metric is relative and it does not give any guarantee about the absolute performance of the selected routing. Oblivious routing aims for the optimal routing regardless of network demand assuming no knowledge of traffic matrix. In another publication [3], Applegate and Cohen present a routing restoration framework that retains nearly optimal performance on the failed network while minimizing traffic flows that did not traverse the failed parts of the network.
Ben-Ameur and Kerivin [16], have introduced a polyhedral set of demands to capture variation in traffic, applying linear programming techniques to compute an optimal stable routing for all traffic demands within uncertainty set. In order to express the relationship between various origindestination traffic flows, the polyhedral model uses demand polytope to capture the traffic demand behaviour. Demand polytope is a bounded set of demand containing all relevant traffic demand configurations. To simplify the solution, [16] used a conservative cost function. The cost function is a linear combination of maximum utilization of all links that does not occur simultaneously in the network. However their formulation has infinite number of constraints. In [42], Juva analyses the use of robust routing through a combination of traffic matrix estimation and its corresponding estimation error bounds, in order to shrink the uncertainty set. The main drawback of stable routing is its inherent dependence on the accurate definition of uncertainty set: on one hand, larger set allows handling a broader group of traffic demands, but at the cost of routing inefficiency. On the other hand, tighter sets produce a more efficient routing scheme but result in poor performance guarantees.

In [71], Zheng Ma et al have introduced COPE, an approach to deal with this trade-off in the size of the uncertainty set combining traditional with the oblivious routing approach. COPE optimizes routing for the predicted demands and bounds worst-case performance to ensure acceptable efficiency under unexpected traffic events. The routing configuration in COPE is averaged over 24 hours hence lacks the adaptability (losing performance efficiency) for real time traffic on smaller time scale (e.g. hourly traffic variations).

An extreme case is presented in [63], where routing is optimized for a single estimated TM and then it is applied for long-term periods (e.g. 24 hours). Traffic uncertainty is characterized by multiple TMs in [75]. In this proposal C. Zhang et al. defines sets of TMs from the previous day, same day of previous week to find optimal routes. In [21], P. Casas and S. Vaton propose a multi-temporal robust routing solution. It provides a fair comparison of stable robust routing and time varying robust routing approach by controlling the demand uncertainty set.

2.2.2 Reactive TE

A completely different approach to deal with uncertainty consists in allowing the network to reconfigure routing in a dynamic fashion when there is a change in traffic demand matrix.

One of the earliest works on distributed and responsive routing algorithm was proposed by Jaffe and Moss [40]. Jaffe and Moss presented a new distributed routing algorithm for the dynamic determination of weighted shortest paths in a network. The dynamic routing problem was intensively discussed for switched telephone networks [6].

In the same direction, there have been proposals for reactive or on line Multipath TE [25, 43]. A distributed method called TeXCP for MPLS traffic engineering was introduced by Khandula et al. [43]. Both balance load in real time, responding to instantaneous traffic demands. The main objective of these reactive algorithms is to avoid network congestion by adaptively balancing the load among paths, based on measurement. Moreover an Adaptive Multipath Routing(AMP) for dynamic traffic engineering was proposed in [33]. However AMP does not consider global network information and restricts the available information to a local scope. A theoretical framework for adaptive traffic engineering is proposed for Distributed Adaptive Traffic Engineering [36].Load balancing is performed over a set of precomputed MPLS paths between ingress and egress routers based on traffic measurement from the network. The authors discussed and prove stability, convergence and optimality. Reactive routing presents a desirable property of keeping routing optimal for dynamic traffic. However, reactive routing algorithms show poor performance under abrupt traffic changes [71]. Despite some benefits, reactive routing is difficult to implement in practice. A Load balacing architecture under changing demands was proposed by [53, 64]. Johansson and Gunnar explored the interplay between estimation of traffic matrix and routing optimization in their Data-driven traffic engineering [41]. In the Data-driven approach [41] Johansson and Gunnar used a measurement driven traffic engineeing to quantify the demand uncertanties for routing optimization.

2.3 Discussion and Summary

On one side, proactive routing is stable but may be costly. On the other side, reactive routing is difficult to implement despite offering several benefits. Moreover, it was proved recently in [22] that this problem is co-NP hard to decide, whether a given network with known capacities can carry each traffic matrix in a demand uncertainty set, when routing is dynamic. This opens a door for us to conduct research in the space between proactive and reactive routing. In a similar effort, we propose a robust routing approach which tries to alleviate the drawbacks of above mentioned approaches. A middle ground between the two methods would be a reasonable demand polytopes around the base TM. We aim to define demand uncertainty set into smaller subsets and compute robust routing on these subsets. Here the obvious question is how many subsets of demand can best map to the possible traffic condition. Surely there are some cases so implausible that we do not need to consider them. We tried to answer this question with simulation experiments and compared with OSPF-Opt to obtain best possible subsets and provide optimal solution in our robust routing approach. In this thesis, we propose a routing approach which is complementary to both proactive and reactive TE.

Our proposal advances in two directions. First, it extends the notion of robust routing under dynamic traffic demand due to DDoS and worms. Second, it provides a single routing configuration for both stable traffic (24 hours) and abrupt traffic (60min). We show the comparison among the different solutions in Table 2.1

Algorithm	No. of TM	Constraints	LP size	Routing
		model		formulation
Applegate and Co-	Infinite	Pipe	finite	Link Based
hen				
Azar et al.	Infinite	No con-	Infinite	Link Based
		straints		
Zhang et al	Finite	NA	Finite	Link Based
Ben-Ameur et al.	Infinite	Hose and	Infinite	Path Based
		Pipe		
Our approach	Finite	Hose and	Finite	Path based
		Pipe		

Table 2.1: Summary of related research work



Robust Routing Problem

3.1 Overview

Many real telecommunication and transportation problems may be represented mathematically in terms of shortest path problems with weights, associated with a cost function. In this chapter we first consider two related problems of maximum flow routing and shortest path routing to gain understanding in terms of flow variables throughout the thesis. We also include in this chapter a definition of the class of network flow problem called multi commodity flow problem with the goal of minimizing the maximum link utilization. In the context of networking, commodity corresponds to demands or supplies between pair of nodes. Our goal in the thesis is to minimize the maximum link utilization. In other words, we define a routing solution so that load can be balanced among shortest routes between number of demand pairs.We use a three node network example(see Figure 3.1 later in this chapter) to understand the Multi commodity flow problem [48].

3.2 Maximal flow routing problem

Consider a network with N nodes and A arcs, through which a single commodity will flow. We associate with each arc (i, j) a lower bound on flow of $l_{ij} = 0$ and an upper bound on flow of c_{ij} . We shall assume that the upper bounds c_{ij} 's are finite integers and are defining the capacity constraint where total flow on arcs should be less than its capacity. There are no costs involved in the maximal flow problem. In such a network, we wish to find the maximum amount of flow from node 1 to node N in the network.

Let f represent the amount of flow in the network from node 1 to node N and r_{ij} represent the flow variable. Then the maximum flow problem may be stated as follows:

$$max f \tag{3.1}$$

subject to

$$\sum_{j=1}^{N} r_{ij} - \sum_{j=1}^{N} r_{ji} = \begin{cases} +f & \text{if } i = 1\\ -f & \text{if } i = N\\ 0 & \text{otherwise} \end{cases}$$
(3.2)

$$r_{ij} \le c_{ij} \ i, j = 1, 2, \dots, N$$
 (3.3)

$$r_{ij} \ge 0 \ i, j = 1, 2, \dots, N$$
 (3.4)

This is called node-arc formulation [48,58] for the maximal flow problem since the constraint matrix is a node-arc incidence matrix, where the sums and inequalities are taken over existing arcs in the network.

3.3 Shortest path routing problem

Suppose that we are given a network graph G with N nodes and A arcs, a non-negative cost C_{ij} , associated with each arc $(i, j) \in A$. The shortest path problem is to find a shortest (or least cost) path from node 1 to node N in graph G. The cost of the path is the sum of the costs on the arcs in the path.

If we set up a network, the shortest path problem is to send a single unit of flow from node 1 to node N at minimal cost. Let b_i defines the vector of supply (if $b_i > 1$), demand (if $b_i < 1$) and $b_i = 0$ for $i \neq 1$ or N. The mathematical formulation of shortest path routing becomes:

$$\min\sum_{i=1}^{N}\sum_{j=1}^{N}C_{ij}r_{ij}$$
(3.5)

subject to

$$\sum_{j=1}^{N} r_{ij} - \sum_{j=1}^{N} r_{ji} = \begin{cases} +f & \text{if } i = 1\\ -f & \text{if } i = N\\ 0 & \text{otherwise} \end{cases}$$
(3.6)

Where $r_{ij} = 0$ or $1 \ i, j = 1, 2, ..., N$

Where the sums and the 0 - 1 requirement are taken over existing arcs in G. The constraints $r_{ij} = 0$ or 1 indicate that each arc is either in the path or not.

Alternatively, if we replace $r_{ij} = 0$ or 1 by $r_{ij} \ge 0$ and if an optimal solution exists, then simplex method would still obtain an integer solution where the value of each variable is zero or one. We may thus solve the integer program as the following linear program:

$$\min\sum_{i=1}^{N}\sum_{j=1}^{N}C_{ij}r_{ij}$$
(3.7)

subject to

$$\sum_{j=1}^{N} r_{ij} - \sum_{j=1}^{N} r_{ji} = \begin{cases} +f & \text{if } i = 1\\ -f & \text{if } i = N\\ 0 & \text{otherwise} \end{cases}$$
(3.8)

Where $r_{ij} \ge 0$ i, j = 1, 2, ..., N

3.4 Robust shortest path problem

As described above, the shortest path problems are the important problems in a range of applications from logistics to telecommunication applications.We can define here the robust counterpart of network application when arc costs are estimated and subjected to uncertainty in the traffic demand data. Mathematical formulation of the robust shortest path routing is as follows:

$$\min\sum_{(i,j)\in A} C_{ij}r_{ij} + \max_{S|S\subseteq A, |s|=\tau} \sum_{(i,j)\in S} d_{ij}r_{ij}$$
(3.9)

subject to

$$\sum_{j:(i,j)\in A} r_{ij} - \sum_{j:(j,i)\in A} r_{ji} = \begin{cases} +1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}$$
(3.10)

3.5 Multi commodity flow problem

In this section, we define a class of network flow problem called multi commodity flow problem in which it is necessary to distinguish the flows in the network. Consider a simple example where there are three commodities that flows through the network. The origin of commodity 1 is node 1, and the destination for commodity 1 is node 3. That is, the commodity 1 must originate at node 1 and terminate only at node 3. Similarly, the origin and destination for commodity 2 be nodes 2 and 1 respectively. Finally, origin and destination for commodity 3 be nodes 3 and 2 respectively. For the given three node network, the multi commodity flow problem is to find a maximum sum of flows in the network $r_1 + r_2 + r_3$, subjected to a constraint that the sum of all commodities flowing on an arc should not exceed the arc capacity $c_{ij} = 1$. Finding a maximal flow path for the three commodity problem described here is relatively easy as there is only path that each commodity can follow on the way from its origin to destination. The paths for commodity 1, 2 and 3 respectively are:

 $P_1 = (1, 2), (2, 3)$ $P_2 = (2, 3), (3, 1)$ $P_3 = (3, 1), (1, 2)$

If we allocate single unit of flow on any of the paths, then the other paths must have zero flow and thus the total flow would be 1+0+0 = 1, to meet the capacity constraint. However, if do not require integer flows then we can provide a better alternative. Suppose that we allocate 0.5 unit of



Figure 3.1: Multi commodity flow problem

flow of each commodity 1,2 and on path P_1,P_2 and P_3 respectively. In this case none of the arc capacities are violated and the total flow of all commodities is 3/2. From this we can see that multi commodity flow problem must essentially solve the constrained routing flow problem amongst multiple commodities.



Routing Problem with Robustness to Traffic Demand

4.1 Motivation

Large IP networks are currently facing a challenging problem of routing under changing traffic conditions. Traffic engineering is playing an important role in optimizing the traffic between ingress (source or origin) and egress (sink or destination) router pairs in the core network. Traffic demand is stable most of the time but there are times when traffic demand is highly dynamic and unpredictable [65]. In the last five years Internet has experienced a tremendous growth with diverse range of applications e.g. voice over IP, video on demand, real time mission critical applications and multimedia streaming services. Classical routing solutions that are based on modeling traffic as single or multiple traffic matrices for dimensioning the network may not provide desired quality of service due to unbalanced load on the network caused by traffic demand variations. In recent years routing *robustness* has drawn the attention of research community [7, 15–17, 42] in the field of network engineering. Intuitively routing robustness is defined as network resilience against changes in the different parameters of the network due to uncertainties.

Reasons for traffic variation: There are several reasons for the traffic variations within the core of ISP network [46]. The actual distribution of traffic between an ingress router and various egress routers is unpredictable due to difficulty in accurately measuring the traffic demand. The intrinsic variation in the traffic demand may be caused by the sudden appearance of flash-crowds responding to international events, denial of service attacks, outbreak of internet worms and viruses. We can list major changes that may affect network performance as below:

- 1. Network topology and connectivity: This includes changes in the capacity of links and failures of links.
- 2. Community of interest (number of active source-destination pairs).
- 3. Traffic demand matrix.
- 4. A sudden and voluminous variation in traffic due to distributed denial of service attacks (DDoS), internet worms and viruses.

Uncertainty in the traffic demand leads us to address some of the potential routing issues:

- 1. Find a robust routing solution over a range of traffic demand matrices.
- 2. Design a traffic demand model to capture the sudden traffic variations.
- 3. Optimize the routing design with robustness to the traffic demand

to provide an efficient routing solution for the network.

In this chapter we study the network routing problem with uncertainty in the traffic demand. We study the performance of our routing solution by evaluating the maximum link utilization and compare the simulation results with OSPF routing.

Rest of the chapter is organized as follows. In the next section, we define the terms such as routing, optimal routing, and feasible routing using routing variables. In section 4.3, we present the motivation. We describe oblivious routing as a major source of motivation. In section 4.4, we present robust routing design. This essentially consists of problem statement, outline of our routing model and the RRT algorithm. Section 4.5 shows performance evaluation of RRT on the real network topologies and synthetic topologies. In this section we present information about the data and the methods of traffic matrix generation used in our simulation experiments.In section 4.6 we present simulation results. Finally we present discussion followed by summary in section 4.7 and 4.8 respectively.

4.2 Notations and symbols

Consider a network topology defined by an undirected graph G = (V, E). Edges $i, j \in E$ are referred to as links. For each link i, j the directed pairs i, j and j, i are called the arcs of G. We denote the set of arcs of G by A. Each link is assigned a capacity c_{ij} which is available for the total flow on i, j in both directions. Consider a set of directed origin-destination pairs, $Q = (s,t) : s, t \in V, s \neq t$. An origin-destination pair (OD) is an oriented pair (s,t) of nodes in V requesting an amount of flow d_{st} from node s to t. Let Q be the set of OD pairs. A traffic matrix $d_k = d_{st}, (s,t) \in Q$ is a vector of all flow requests between node i and node j.

Routing Variable:Routing variable or flow variable is defined as fraction of (OD) demand flowing on edge i, j in the direction $s \to t$ and is denoted as r_{ij}^{st} .

Multipath routing: When there are multiple shortest paths between OD pairs, then routing can be modeled as a routing vector r_{ij}^{st} over K shortest paths. We use K-shortest path routing algorithm in our implementation. The set of paths for each OD pairs $s \to t$ is defined as $P = P^1, P^2, \dots, P^K$. Led as a routing vector r_{ij}^{st} over K shortest paths. We use K-shortest path routing algorithm in our implementation. The set of paths for each OD pairs $s \to t$ is defined as $P = P^1, P^2, \dots, P^K$. Led as a routing vector r_{ij}^{st} over K shortest paths. We use K-shortest path routing algorithm in our implementation. The set of paths for each OD pairs $s \to t$ is defined as $P = P^1, P^2, \dots, P^K$.

A multi path routing may be defined as a fraction of traffic demand d_{st} over path set P^K as follows:

$$\sum_{j:i,j\in E} r_{ij}^{st} - r_{ji}^{st} = \begin{cases} +1 & \text{if } i = s \\ -1 & \text{if } j = t \\ 0 & \text{otherwise} \end{cases}$$
(4.1)

$$\sum_{k} r_{ij}^{k} = 1, r_{ij}^{k} \ge 0 \tag{4.2}$$

Feasible routing: Let us define set of routes on G as Λ . Given the traffic demand d_k , we may define routing as feasible i.e. $r \in \Lambda$ w.r.t. d_k if the capacity of none of the links is overloaded that is if the following inequality is satisfied:

$$\sum_{s,t\in Q} d_{st}(r_{ij}^{st} + r_{ji}^{st}) \le c_{ij}, \, i, j \in E$$

Total Load: When the traffic is flowing in both directions of the network then the total load can be expressed mathematically as follows:

$$Load(e, f, d) = \sum_{(s,t) \in Q} d_{st} (r_{ij}^{st} + r_{ji}^{st})$$

Maximum link utilization: We use maximum link utilization as an objective function in this chapter, which may be defined as a maximum of the ratio of the link load and total link capacity.

$$MaxU_d = max_{(i,j)\in E} \frac{Load(e,f,d)}{c_{ij}}$$

Optimal routing: We may now define the optimal routing as the minimization of maximum link utilization $MaxU_d$ as follows:

 $OptMaxU_d = min_{r \in \Lambda}MaxU_d$ The routing problem with minimum $MaxU_d$ for a fixed demand d_k is modeled as:led as:

$$min t \tag{4.3}$$

subject to

$$\sum_{s,t\in Q} \frac{d_{st}(r_{ij}^{st} + r_{ji}^{st})}{c_{ij}} \le t, \forall i, j \in E$$

$$(4.4)$$

 $r\in\Lambda$

Where r is feasible routing. The variable t defines the optimal value for the maximum link utilization. This provides a routing solution with minimization of maximum link utilization which is a useful network performance measurement metric.

4.3 Oblivious routing

The concept of oblivious routing [5] aims at developing a robust routing algorithm basing routing decisions only on local knowledge and therefore can be deployed efficiently in a distributed environment. Traditionally, for an oblivious routing algorithm the routing path between an origindestination pair (s, t) may depend only on s and t. Our focus is a routing algorithm that aim to minimize the congestion, which is defined as Maximum Link Utilization of a network arc.

We consider a robust routing problem when the traffic demands between node-pairs are chosen as per the gravity model (see the data section of this chapter). Routing scheme consists of allocating fraction of flow from s to t for every node-pair (s,t). The flow for each pair (s,t) determines how the demand from s to t is routed.

The goal of oblivious routing is to minimize the edge-congestion which is defined below. For a given traffic-matrix D and a given routing algorithm, we define the total load of an edge as the amount of flow routed along this edge. Relative load or Link Utilization is defined as the total load of an edge divided by its capacity. The congestion is defined as Maximum Link Utilization of an edge. Oblivious routing can be viewed as providing robust routing solution for a class of traffic demand matrices.

We define $MaxU_d$ to be the maximum link utilization of the routing guided by the routing flows for traffic-matrix D, in which each path from s to t for a commodity pair (s, t) gets flow proportional to its share as per the routing mode. These modes are defined in the RRT pseudo-code in the following section.

Symbol	Description
d_{ij}	Traffic demand between each ingress node i and each egress node
	$j, (i, j \in N)$
r^a_{ij}	Fraction of traffic demand from i to j carried through an arc $a \in A$
r_{ij}^p	Fraction of traffic demand from i to j carried through the path
	$p \in P(i,j)$
c_a	Capacity of each arc
$MaxU_d$	Maximum link utilization

Table 4.1: Notations used in the problem formulation

Let $OPTMaxU_d$ be the optimal value of Maximum Link Utilization for a traffic-matrix D which can be obtained by solving a linear program (LP) as shown by equation 4.1.

4.4 Robust routing design

4.4.1 Mathematical formulation

In this section we present the multi commodity flow problem with an objective function as Maximum Link Utilization. We denote each entry of traffic profile as a traffic demand between an IE pair. We use Linear Programming (LP) to solve the equations 4.3 and 4.4, as it is a polynomial-time formulation. We use following notation to formulate the problem mathematically:

4.4.2 Problem statement

We consider Maximum Link Utilization (MLU) as the performance metric. For a given fraction of traffic demand r_{ij}^a and demand d_{ij} , the Maximum Link Utilization can be defined as the maximum ratio between link load and link capacity. Routing optimization [62] consists of minimizing this MLU associated with the traffic demand D. There are other performance metrics that one might consider instead of MLU, such as cost and end-to-end path delay. Since our main focus is to avoid congestion in the network, hence MLU is the natural choice to evaluate the network performance.

$$min \ MaxU_d$$
 (4.5)

subject to

$$\sum_{p \in P(i,j)} r_{ij}^p \ge 1 \ \forall i, j \in N$$

$$(4.6)$$

$$\sum_{p \in P(i,j)} r_{ij}^p \le r_{ij}^a \forall i, j \in N, \forall a \in A$$
(4.7)

$$\sum_{i,j\in N} d_{ij} r_{ij}^a \le Max U_d. c_a \forall d_{ij} \in D, \forall a \in A$$
(4.8)

$$r_{ij}^p \ge 0 \ \forall p \in P(i,j), \forall i,j \in N$$
(4.9)

$$r_{ij}^a \ge 0 \ \forall a \in A, \forall i, j \in N$$

$$(4.10)$$

This formulation is based on arc-path flow formulation. The constraint in inequality 4.6 refers to the multi path routing as traffic demand between each pair of nodes may split among multiple paths. Flow variable for each is defined by an inequality 4.7. Total traffic flowing through each arc must be less than the capacity times the MLU, defined by constraint in 4.8. The last two sets of constraints must be satisfied to ensure the positivity of flow variables r_{ij}^p and r_{ij}^a .

The problem of routing traffic demands to minimize congestion over mul-

tiple paths is NP- hard [4]. Thus we use the local search heuristic proposed by Fortz and Thorup [30]. Given a network topology, link capacity and demand matrix, the local search heuristic evaluates the points in a search space where a point is represented by a set of weights. We aim to solve our Linear Program (LP) to get an optimal multi-path solution for each traffic demand. After solving the LP, path decomposition results in reduced set of paths for each traffic demand. Each path having a value assigned to it that represents the fraction of the traffic demand being routed through the path.

4.4.3 Network Model

Traffic Engineering refers to the optimization of network configuration under given network and traffic conditions. In order to find a robust routing, a number of steps are needed. Firstly we need to collect the information about network topology and current traffic situation. Robust TE algorithm needs as an input a traffic demand matrix between each pair of nodes in the network. Obtaining the traffic matrix in a large backbone network is a challenging task; hence we are using the available data from Rocketfuel [65]. The traffic matrix with network topology and link capacities are used as input to the optimization of the routing.

The network model to implement the robust TE block is shown in the Figure 4.1. RRT is using topologies graph i.e. (N, A) as input and generating maximum link utilization as output. The framework of routing algorithm shown in the following Figure 4.1:



Figure 4.1: Schematic diagram for the robust routing model

4.4.4 Algorithm

Outline of RRT algorithm is shown in Table: 4.2. We use Dijkstra's algorithm to compute the initial path set P and ECMP-KSP to obtain the Kshortest paths. In the next step we use LP to optimize the MLU, subject to the constraints defined in the equations 4.5-4.10. We use ILOG-CPLEX 10.2 [39] to solve the LP. For path update we use column generation approach. This column generation procedure will only compute and add the subset of paths that can reduce the value of Maximum Link Utilization at each step. A master program solving each sub problem based on the constraints, will prove optimality of current solution. Our goal is to focus on a routing solution with robustness to the traffic demand.

Robust-Routing () {

$$TM = \text{ReadTrafficMatrix}(\text{input File})$$

 $W = \text{ReadWeightFromGraph}(G')$
for each $(s, t, c_a) \subset TM$ {
switch (Path-Selection-Scheme)
case(Path-Selection-Scheme == WSP)
 $P = \text{Dijkstra}(s, t, G), P$ is one path
case(Path-Selection-Scheme ===ECMP)
 $P = \text{ecmp}(s, t, G), P$ is set of paths
}
AllocateBandwidthToGraph (P, G)
}
 $MaxU_d = max \frac{linkload}{totallinkcapacity}$ }
ECMP Procedure () {
ecmp (s, t, G) {
 $\aleph = k_{SP}(s, t, G, k)$
 $P = 0$
 $Q = Q' = 0$
 $K = first \in \aleph$
While $(K! == 0)$ {
 $Q = K$
if((weight $\in Q == weight \in Q')||(Q' == 0))$ {
 $P = P \bigcup Q$
 $Q' = Q$
}
else
break
 $K = K \rightarrow \text{next}$
}end of while loop
return P
}



4.5 Performance evaluation

4.5.1 Data

We describe in this section the topologies used in our robust routing test bed. We use the real network benchmark data provided by Rocketfuel project in the public domain.Rocketfuel developed a set of measurements and released publicly available approximate router-level topologies of representative collection of Internet Service Providers.

4.5.2 Topologies

The topologies provided by Rocketfuel project did not include the capacities of the links, which are needed in our experimentation. We use four ISP topologies from the Rocketfuel project dataset with capacities assigned on each link by [43]. These link capacities are in line with ISPs. For simulation the core to core (CC) links have high capacity (10Gbps) and other links (Point to Point i.e. PP and Core to Point i.e. CP) have smaller capacities (2.5Gbps). We also consider the pure random topologies generated by GT-ITM [35] to include larger traffic matrices to validate our results.

4.5.3 Methods of traffic matrices(TMs) generation

We use two different methods for synthetic traffic matrix generation, which we refer as Gravity TMs and Bimodal TM.

* Gravity TMs: Similar to [4, 43] we use the gravity model of traf-

fic matrix generation for our robust routing test bed. The gravity model is based on Newton's gravitation law to estimate the traffic demand between pair of nodes in the network. According to the gravity model, the total traffic demand between the Origin-Destination (OD) pair of nodes is proportional to the product of the amount of traffic flow.

* Bimodal TM: We use Bimodal TM generation method to generate the random traffic matrix to test it with GT-ITM topologies. The random bimodal distribution samples randomly a fraction of OD pairs and then assigns a demand for the pair uniformly at random.

4.5.4 Simulation set up

In order to evaluate the effectiveness of our approach, we conducted our simulation on artificially generated random topologies using GT-ITM as well as on actual ISP topologies from the Rocketfuel project [36]. We use five different topologies to test out algorithms. We use AS1221 (Telstra, Australia), AS1239 (Sprint, US), AS6461 (Abovenet, US), AS1755 (Ebone, Europe), AS3967 (Exodus, Europe). We are using following routing techniques for simulation:

* Optimal OSPF: We use OSPFOpt [43] for preliminary experiments to evaluate the network performance. OSPF-Opt is an implementation of search algorithms described in Fortz and Thorup's paper [31].Given a traffic demand matrix and topology OSPF-Opt computes a set of weights which when used with OSPF protocol result in low cost or low maximum link utilization. * RRT: We conducted our simulations using ns-2 [55].We use Stanford Graph Base libraries (SGB) [45] within ns-2 to simulate the RRT and to provide sets of paths corresponding to compute link weights and traffic traces under changing traffic demand conditions. RRT uses three routing implementations. Equal cost multipath (ECMP), Weighted Shortest Path (WSP) and K- shortest path (KSET) algorithms are implemented in RRT for traffic allocation. Maximum Link Utilization is calculated based on both OSPF and RRT. K-shortest path routing mode is using K = 10 or 20 in order to evaluate the performance.

4.6 Result and analysis

4.6.1 Robust routing environment on set of traffic matrices

In this case, we perform the simulation experiments using three routing modes in RRT algorithm and compare the performance with Optimal OSPF. Objective of our simulation is to minimize the Maximum Link Utilization.

We perform simulation on five real network ISP topologies. In Figure 4.2 we observe that Maximum Link Utilization approaches nearly 1 for AS1239 topology, which necessitates the need of traffic engineering. In this case, we observe that the performance of K-shortest path routing implementation in RRT is relatively better to OSPF-Opt for the ISP topologies AS1221, AS6461 and AS3967. RRT-KSET outperforms the



Figure 4.2: Performance evaluation on real network topologies. The value on the y axis denotes MLU, which is expressed as a fraction instead of percentage. Link weights are assigned as per inverse capacity rule and the traffic margin is set to 1.1

other routing modes for these topology instances as well. On the contrary, RRT-ECMP and RRT-WSP routing algorithms result in improved MLU for the AS1221 and AS1239 topologies as compared to OSPF-Opt.

It is seen in Figure 4.3 that when traffic margin is assigned a value 5.0, RRT-KSET performs better resulting in improved MLU compared to other methods for the topologies AS1221, AS6461 and AS3967. Whereas RRT-ECMP, i.e. equal cost multipath method for traffic allocation results in similar MLU as OSPF-Opt for the other topologies AS1239 and AS1755. We have included Widest Shortest Path routing algorithm in RRT as well.

Moreover RRT performance provides mixed results on assigning traffic margin 10 as shown in Figure 4.4 OSPF_Opt performs better compared to RRT for AS1239, AS1755 and AS3967. RRT shows comparable performance for AS6461. RRT results in significantly improved MLU for



Figure 4.3: Performance evaluation on real network topologies. The value on the y axis denotes MLU, which is expressed as a fraction instead of percentage. Link weights are assigned as inverse capacity and traffic margin is set to 5.0



Figure 4.4: Performance evaluation on real network topologies. The value on the y axis denotes MLU, which is expressed as a fraction instead of percentage. Link weights are assigned as per inverse capacity rule and traffic margin is set to 10.0



Figure 4.5: Performance evaluation on real network topologies. The value on the y axis denotes MLU, which is expressed as a fraction instead of percentage. Link weights are assigned randomly and traffic margin is set to 5.0

AS1221 as compared to OSPF-Opt.

We perform additional simulation on the same topologies with weights randomly generated and observed that RRT-KSET performs better as compared to RRT-ECMP, RRT-WSP for all five topologies and OSPF-Opt for AS1221.

4.6.2 Routing performance on larger random topologies

Case (i): Performance evaluation on pure random graphs

Furthermore, the result in Figure 4.6 shows the performance of RRT algorithms on pure random graphs generated by GT-ITM. We observe that the K-shortest path routing mode in RRT outperforms the other two modes on graphs r10, r30 and r50 resulting in improved Maximum Link Utilization. On the other hand, RRT-ECMP proves better for



Figure 4.6: Maximum Link Utilization of RRT algorithms on GT-ITM using pure random graphs

Topology	RRT- ECMP	RRT-WSP	RRT-KSET
r10	0.3857	0.4159	0.3537
r30	0.8304	0.8433	0.8304
r50	0.4558	0.7279	0.4714
r100	0.4084	0.9488	0.5170

Table 4.3: Maximum Link Utilization of RRT algorithms on GT-ITM using pure random graphs

r100.Overall, we found that the K-shortest path RRT performs better as compared to other routing modes on range of random graphs.

Figure 4.7, 4.8 and 4.9 show the link utilization for the three routing modes implemented in RRT on pure random graphs of GT-ITM. We observe that the RRT-SET has comparatively even link utilization as compared to the other two modes i.e. RRT-ECMP and RRT-WSP. RRT-WSP shows poor link utilization pattern when number of nodes are 50(r50 topology).

Case (ii): Performance evaluation on transit-stub graphs

In the previous case we consider the pure random graphs. Pure random



Figure 4.7: Link Utilization on GT-ITM random graph r10 (number of nodes =10)



Figure 4.8: Link Utilization on GT-ITM random graph r30 (number of nodes =30)



Figure 4.9: Link Utilization on GT-ITM random graph r50 (number of nodes =50)

Topology	RRT-	RRT-WSP	RRT-KSET
	ECMP		
Ts100a	0.3358	0.3358	0.3358
Ts100b	0.2778	0.2616	0.2718
Ts100c	0.0566	0.0676	0.0592

Table 4.4: Maximum Link Utilization on GT-ITM using transit-stub graphs (number of nodes N = 100)

graphs are easier to use in terms of studying network simulation. On the other hand pure random graphs do not closely resemble to the real internetwork. In this section we consider graphs generated by GT-ITM using transit-stub topology format.Transit-stub graphs may capture the interdomain traffic engineering. The traffic is generated using bimodal method of traffic generation. We use three transit-stub topology instances Ts100a, Ts100b and Ts100c. We define the link capacities as follows: C-C: 10Gb, C-P: 2.5Gb, and P-P 1.5Gb. The link weights are assigned as per inverse capacity rule. Each Ts100 represents a transit-stub graph with number of nodes assigned as 100. In order to examine the performance of RRT routing modes, we assign extra stub-stub and stub-transit links, by considering 35% of total number of links.Table 3.2 shows that RRT-WSP performs better in for Ts100a and Ts100b, whereas performance of RRT-KSET is better for Ts100c.

4.7 Discussion

In the initial phase of simulation experiments, we use real network topologies to validate the performance of three routing modes: ECMP, WSP and K-SET. We use traffic margin to provide traffic demand matrix as per the link load. We consider three different margins 1.1, 5.0 and 10.0. It can be seen from the above results that RRT produce nearly similar value for maximum link utilization as compared to OSPF-Opt for the topologies used in experimentation. We also notice that among all routing modes in RRT, RRT-KSET: K-Shortest implementation outperforms the other routing methods for the topologies AS1221, AS6461 and AS3967. However, it gives the inference that, in essence, MLU increases when the link utilization increases and when the traffic increases.

We have observed through the simulation results shown in Figure 4.2 - 4.5 that network congestion increases when the traffic demand increases. This is more clear for the larger topologies AS1239 and AS3967 when the value of maximum link utilization approaches to 1. It is seen from the results that the value of maximum link utilization with K-Shortest path version of RRT (RRT-KSET) is comparatively better for smaller topology(AS1221) as well as larger topologies(AS1239 and AS3967).

In our simulation, we also consider the pure random graphs. Pure random graphs are easier to use in terms of studying network simulation. On the other hand pure random networks do not closely resemble real internetwork. Indeed, it is interesting to test the effectiveness of RRT on larger random graphs. It can be seen from the results, that we obtain reasonable load balancing with RRT routing modes: RRT-ECMP and RRT-KSET except for topology r100 when RRT-KSET gives MLU that approaches to 1.However, it is beyond the scope of this thesis to see the effect of the nature of traffic for pure random graphs as the traffic is generated using Bimodal method. Furthermore, we use transit-stub graphs to test the interplay of interdomain traffic engineering. We use a larger network with 100 nodes and use three different transit - stub graphs. It is interesting to note that observe that the performance of RRT is significantly better for transit stub graphs resulting in low value for maximum link utilization (around 0.3).

4.8 Summary

In this chapter, we study the routing robustness to the changes in traffic demand and proposed an approach for routing of flows in the IP networks. The essence of our work is based on determining the maximum link utilization for each link to the changes in the traffic demand of a network. We gave a mathematical model for the links and used a local search heuristic to reflect the changes in the demands.Our algorithm RRT computes the maximum link utilization to measure the performance of a network. The simulation results show the performance of RRT compared to OSPF-TE on range of traffic data: real network topologies and random topologies. We compare our algorithm with OSPF-Opt in the first phase of simulation experiments and found that the RRT performs better for most of the topologies.Furthermore, we test RRT performance on pure random and transit-stub graphs.

In a similar study done in [5,12], it is observed that the network congestion may be affected by the variability in the traffic demand. Usually Internet traffic is assumed stationary and routing solutions become oblivious to cope against the changes in the traffic demand conditions and this is well addressed in the above studies. We conduct the simulation in a more specific way.

As expected, from the nature of Internet traffic the congestion might be a more severe problem for highly variable demand intervals.Hence a detailed robust performance evaluation under dynamic traffic might be useful as an extension to the existing solution.There are many issues that remain to be investigated to improve the proposed robust routing in this chapter. We believe that the idea of robust routing in this chapter can be improved for the dynamic traffic demand. We explore this in detail in the next chapter. Yet another research challenge is to look for a routing algorithm that can compromise between the of cost of routing robustness versus robust optimization of maximum Link utilization. Chapter 5

Robust Routing Under Dynamic Traffic Demand

5.1 Introduction

In this chapter, we define a routing algorithm as robust if it can cope with sudden changes due to unusual variations in the internet traffic such as DDoS and worms. We aim to minimize the congestion on an edge of a network on a set of traffic matrices. Furthermore, we compute a congestion ratio, using an uncertainty parameter to capture dynamic variations. In the rest of the paper congestion is measured by maximum link utilization.

Robust network design under variability in the traffic demand has been a topic of research investigation in past few years [14,21,44,47,69].Duffield et al. [24,57] introduced an uncertainty model that allows all demand within the upper and lower bound for the incoming and outgoing links. Bertsimas and Sim [19] define an uncertainty demand model where all demands are allowed to take an upper or lower value, while imposing the maximum number of demands that may attain the maximum value simultaneously which corresponds to the network congestion.

Ben-Ameur and Kerivin [16] present an uncertainty model, whereby traffic demands are defined as a set of demand polyhedron. They formulate the robust routing optimization problem with polyhedron uncertainty as a semi-infinite linear program where all demands within polyhedron are associated with a constraint in the linear program. In order to tackle with infinite number of constraints they propose an efficient iterative routing solution that considers an initial set of traffic demands. At each step of master LP program, a separation problem is solved to generate the unroutable traffic demand.

Other problems are aiming to minimize the maximum utilization of the critical links in the network. Applegate and Cohen [5] propose oblivious routing problem with fairly limited knowledge of traffic demands. They also argue that all demands that admits feasible routing, constrained by network capacity, are possible. In the same direction, Belloti and Pinar [12] study a routing problem with upper and lower bound on demands defined as box uncertainty and ellipsoidal uncertainty of traffic demand in which mean-covariance information on demands are available.

Our work is complementary to [5,12,13]. We consider the robust routing so that problem can be solved efficiently. Then we define a demand polytope to define the traffic variation caused by uncertainties e.g. DDoS and worms. We aim to develop a simple robust routing solution on a family of real network and random topologies to minimize the maximum Link
Utilization to improve the existing framework.

The structure of the rest of the chapter is organized as follows: We present the performance metric under demand uncertainty in the next section. In section 5.3, we define the optimal routing problem under dynamic traffic demand. In section 5.4, we present the motivation of using traffic models. This includes, hose model, polyhedral model and discrete model or Bertsimas and Sim model. In section 5.5, we present the routing algorithm followed by the mathematical formulation of the robust routing problems. In section 5.6, we evaluate the performance and show simulation results. This includes the comparison of robust routing with optimal on real and random topologies. Finally, in section 5.7, we present our discussion followed by the conclusion in section 5.8.

5.2 Performance metric under demand uncertainty

When the routing problem is solved under demand uncertainty, the performance is measured by congestion ratio or the maximum link utilization ratio. In the previous chapter congestion is defined as maximum link utilization (see section 4.3).For a given routing r and set of routable demands S(D), the congestion ratio is defined as the ratio of the maximum link utilization of the routing r to the maximum link utilization of the optimal routing for S(D). The congestion ratio measures, how far the routing r

Symbol	Description
D	set of all possible demands
r	feasible routing
S(D)	set of routable demands
Q	Directed origin-destination pair
$OptMaxU_{d}$	Optimal maximum link utilization
z	variable defining congestion ratio
w	variable for optimal maximum link utilization
d_k	Base traffic matrix between pair ij
B_{in}, B_{out}	Bounds for incoming and outgoing traffic

Table 5.1: Notations used in the problem formulation

is from optimal routing for a set of routable demand S(D).

$$CR(r, S(D)) = \frac{MaxU_d}{OptMaxU_d}$$
(5.1)

The congestion ratio(CR) is usually greater than 1 in practice. In other words, it is equal to 1, when routing r is an optimal routing for S(D). The worst case congestion ratio is the maximum of all congestion ratio, when the demand set includes all possible demands i.e. D for a given r. We use congestion ratio to measure the performance of robust routing framework in this chapter.

5.3 Optimal routing under dynamic traffic demand

Consider a network topology defined by an undirected graph G = (V, E). Edges $i, j \in E$ are referred to as links. For each link i, j, the directed pairs i, j and j, i are called the arcs of G. We denote the set of arcs of G by A. Each link is assigned a capacity c_{ij} , which is available for the total flow on i, j in both directions. Consider a set of directed origin-destination pairs, $Q = (s,t) : s, t \in V, s \neq t$ An origin-destination pair (OD) is an oriented pair (s,t) of nodes in V requesting an amount of flow d_{st} to send from node s to t. Let Q be the set of (OD) pairs. A traffic matrix $d_k =$ $d_{st(s,t)\in Q}$ is a vector of all flow requests between node i and node j.

Now consider the case when traffic demand is not fixed and is not known and a set D of all possible matrices are given. Problem is now to find the best routing configuration for all demands in set D. Let's use a new demand set of routable demands as S(D). In case of uncertain demand we consider a worst case approach where routing r is measured by congestion ratio, over larger set of demands $D \in S(D)$

$$minz$$
 (5.2)

subject to

$$z \ge \max_{D \in S(D)} \sum \frac{d_{st}(r_{ij}^{st} + r_{ji}^{st})/c_{ij}}{OptMaxU_d}, \forall i, j \in E, \forall d_k \in D$$
(5.3)

 $r \in \Lambda$

Here z is a variable defining minimization of maximum link utilization(the quantity after summation). The LP in the above equation may be rewritten using duality theorem as follows:

$$\max_{D \in S(D)} \sum_{(s,t) \in Q} d_{st} (r_{ij}^{st} + r_{ji}^{st} - zc_{ij}OptMaxU_d) \le 0, \forall i, j \in E$$
 (5.4)

5.4 Traffic model

We are motivated by the popular traffic demand models used to model the uncertainties in the traffic demand. We considered two traffic demand models to specify the traffic demand matrix d_k to test our model and algorithm.

5.4.1 Hose model

This uncertainty model was introduced by Duffield et al [24]. The model is inspired by the data networks in which users have fixed-capacity connection to the Internet. Here the set of traffic demand is defined by the bounds on the total flow (in units of capacity) between each pair of terminal nodes. Given this information, the traffic demand matrix D can be considered to be the set of all traffic demand matrices with respect to capacity constraints at each node i, then for the symmetric hose model the demand can be defined with the following bound: $\sum_{j\neq i} d_k \leq B, \forall i \in V$ When the upload and download link capacities are different e.g. in VPN network [2], then traffic demand may have two separate bounds for the incoming and outgoing traffic. That is for each node i there are two nonnegative bounds B_{in} and B_{out} respectively for $d_k \in D$ iff: $d_k \leq B_{out}$ and $d_k \leq B_{in}$

5.4.2 Polyhedral model

Proposed by Ben Ameur and Kerivin [16], Polyhedral model of traffic assume that traffic demand between node pairs can be carried on multiple paths. This model does not consider any probabilistic assumption about the traffic demands. There are two ways to define the polyhedron.Polyhedron P(D) could be defined as a set of linear inequalities in which case the size of system is part of the input. The polytope D can be defined using Minkowski's theorem as a set of extreme points and can be expressed mathematically as follows:

$$D = d_k \in \Re^{|v||v|-1}; d_k = \sum_{i=1}^r \lambda^i d_k^i, \sum_{i=1}^r \lambda_i = 1, \lambda \ge 0$$
(5.5)

5.4.3 Bertimas and Sim. model

This is basically a discrete model that assumes the lower and upper bound for the pair wise demands. In most network design problems considering this type of bounding under uncertainty means all demand can get their peak values simultaneously. To overcome this, a parameter τ is defined to compromise between the robustness and conservative nature of resulting solution. This is the robust optimization approach defined by Bertsimas and Sim [19].

In the dynamic traffic demand variation problem τ defines the maximum number of demand pairs whose demand would change within their uncertainty limits due to DDoS to affect the solution adversely. Robust-Routing() { Input(Traffic Matrix TM, ReadWeightFromGraph G(W)) for each TM { solve shortest path routing problem } done Compute $MLU = max \frac{linkLoad}{linkCapacity}$ Apply the LP to obtain the Optimal MLU on the set of traffic-matrices DCompute the congestion ratio }

Table 5.2: Routing under dynamic demand

5.5 Robust routing in the face of voluminous traffic demand

5.5.1 Robust algorithm

Given a traffic-matrix and weights, RRT runs three routing algorithms to compute the MLU. First, RRT generates an initial path set P using Widest Shortest Path routing algorithm. Alternatively, RRT-ECMP may be used to allocate flow by evenly splitting the traffic among K number of paths. In order to make RRT more effective, we implemented K-shortest path routing algorithm with weights computed as inverse of link capacity. In the next step we apply linear programming to minimize the MLU. We outline the routing procedure under dynamic demand below:

5.5.2 Mathematical formulation

We give an arc based flow formulation as in [5, 13]. The model includes a penalty term for the critically congested links in the path selection to evaluate the robust routing performance. In the arc based formulation routing variables are defined on the links and flow conservation constraints are defined on the node for each OD pair. The left-hand side in equation 5.4 can be formulated as the following robust routing problem RR1, for each edge $i, j \in E$ with the set of inequalities:

RR1 :

$$\max\sum_{(s,t)\in Q} d_{st}(r_{ij}^{st} + r_{ij}^{st}) - zc_{ij}w^*$$
(5.6)

subject to

$$\sum_{j:(s,t)\in Q} (g_{ij}^{st} - g_{ji}^{st}) = d_{st}, \forall (s,t) \in Q$$
(5.7)

$$\sum_{j:(s,t)\in Q} (g_{ij}^{st} - g_{ji}^{st}) = 0, \forall i \in Vs, t, s, t \in Q$$
(5.8)

$$\sum_{j:(s,t)\in Q} (g_{ij}^{st} - g_{ji}^{st}) \le c_{ij}w^*, \forall i, j \in E$$
(5.9)

$$w^* < 1$$
 (5.10)

$$\sum_{s,t} a_x^{st} d_{st} \le a_x, \forall x = 1, \dots, K$$
(5.11)

$$g_{ij}^{st} \ge 0, \forall (i,j) \in A, (s,t) \in Q$$

$$(5.12)$$

$$d_{st} \ge 0, \forall (s,t) \in Q \tag{5.13}$$

Where the optimal MLU variable is $w^* = OptMaxU_d$. Inequalities in 5.7 and 5.8 are defining flow conservation constraints. They state that, for each commodity, the difference between the flow that enters and the flow that leaves each node is equal to the supply/demand of the same node. Capacity constraint is defined by the inequality 5.9 which states that the total flow on each arc must be bounded by the capacity times

the optimal MLU variable. We consider the bounded demand case, where none of links will be used with full capacity thus implying $w^* < 1$, defined by 5.10. Demand polytope is defined as the cutting plane to model a notion of feasible demands, which admits at least one feasible routing. This is defined by the inequalities 5.11 and 5.13.Flow variable is shown in equation 5.12.

RR2(Dual of RR1):

$$min(\chi_{ij} + \sum_{x=1}^{K} a_x \alpha_x^{ij}) \tag{5.14}$$

subject to

$$\prod_{h,ij}^{st} - \prod_{k,ij}^{st} + \eta_{hk,ij} \ge 0, \forall (h,k) \in A, (s,t) \in Q$$
(5.15)

$$-\pi_{ij}^{st} + \sum_{x=1}^{K} a_x^{st} \lambda_x^{st} \ge r_{ij}^{st} + r_{ji}^{st}, \forall (s,t) \in Q$$
(5.16)

$$-\sum_{h,k} c_{hk} \eta_{hk,ij} + \chi_{ij} = -zc_{ij} \tag{5.17}$$

$$\eta_{hk,ij} \ge 0, \forall h, k \in E \tag{5.18}$$

$$\chi_{ij} \ge 0 \tag{5.19}$$

$$\lambda_x^{ij} \ge 0, \forall x = 1, \dots, K \tag{5.20}$$

In dual LP as shown in equation 5.14 to 5.20, we use primal variables corresponding to the dual constraints. Dual objective function includes dual variable for demand as χ_{ij} . which refers to penalizing for the nonroutable demands. Since we are including non-routable demands as well by considering larger demand set D than the routable demand set S(D). This is defined by the constraints 5.12. In 5.12, we define the most critical demand d for an edge i, j i. e. the demand that makes edge i, j most congested. Computation results are shown in the following section.

5.6 Performance evaluation

5.6.1 Experimental setup

In this section we evaluate the performance of robust routing by performing several experiments on the topologies from Rocketfuel. We have AMPL 10.2 to model our mathematical formulations and CPLEX 10.2 mixed integer programming(MIP) solver. MIP solver in CPLEX is using Branch and Cut with column generation to solve the mathematical program given as an input. We have collected topology information from the Rocketfuel project [65]: AS1221, nsf, AS6461 and AS4755. For each topology, we have number of nodes, number of arcs and weights. As mentioned in the previous chapter, we assume that weights are following the inverse capacity rule: the weight of each link is inversely proportional to its capacity $c_{ij} = \frac{1}{w_{ij}}$. For traffic generation between each pair of nodes, we use Gravity Model as explained in chapter 4. We construct demand polyhedral D using the Gravity model [5, 12]. We conducted our experiments in the following steps:

- * We collected the topologies from the real networks. For each topology the current link weights(w) and information about the number of packets entering and leaving each node are available.
- * We introduce uncertainty in the traffic demand for each element in traffic matrix using uncertainty parameter $p\{1.0, 1.5, 2.0, 2.5, 3.0,$

3.5, 4.0, 4.5, 5.0}. We tested our robust routing procedure where all the entries in the traffic matrix were scaled.

- * We solved an optimal routing problem defined in 5.3 for each topology to minimize maximum link utilization.
- * We then solved robust routing problem formulated in 5.4 to 5.11, with optimal maximum link utilization as an input, to obtain robust routing solution that we defined as congestion ratio.

5.6.2 Demand model under uncertainty due to DDoS and worms

This is basically a discrete model that assumes the lower and upper bound for the pair wise demands. In most network design problems considering this type of bounding under uncertainty means all demand can get their peak values simultaneously. To overcome this, a parameter is defined to compromise between the robustness and conservative nature of resulting solution. This is the robust optimization approach defined by Bertsimas and Sim [19].

In the robust optimization problem τ defines the maximum number of demand pairs whose demand would change within their uncertainty limits due to DDoS to affect the solution adversely. Suppose that you have a set of point-to-point demands 1, 2,, K each from source s_k to destination t_k with k in 1, 2,, K.

Their traffic value can be either d_k or D(k), for instance D(k) is the peak value. If consider the demand uncertainty, not all traffic demands may vary simultaneously. Hence let say τ such demand can have peak value, i.e. D(k). All others have value d_k , where d_k is the base traffic demand. This demand polyhedron in case of uncertainties due to DDoS or worms can be modelled as follows:

$$d_{k} = d_{k} + \alpha(k)(D(k) - d_{k})$$
(5.21)

$$\sum_{k \in K} \alpha(k) \le \tau \tag{5.22}$$

Usually τ is 0.10 to 0.15 of the K. Here the polyhedron is described by all the values of α in [0,1] for which the above constraints hold. We propose this particular model of demand uncertainty for the dynamic traffic demand variation due to DDoS.

5.6.3 Results

We use AMPL to model our LP formulations of robust routing solutions RR1 and RR2. We use Cplex10.2 to solve the mathematical problem and computing optimal MLU, congestion ratio as network performance. In order to conduct our experiments we use ISP topologies of different sizes collected from Rocketfuel project [65]. We use gravity model to generate the demand polyhedron D for all instances. We define the base traffic demand as the product of attraction A and repulsion R terms (as per law of gravitation) scaled by a number *beta* to define the direction for the base traffic demand matrix. We also create variants of each traffic matrix using different value for the uncertainty parameter p(1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0). Larger p values are indicative of higher demand variations resulting in higher value of congestion ratio.



Figure 5.1: Congestion ratio for ISP topology AS1221



Figure 5.2: Congestion ratio for ISP topology AS6461



Figure 5.3: Congestion ratio for ISP topology nsf

AS	р	Robust	Optimal
AS1221	1.0	1.96	1.00
	1.5	1.998	1.00
	2.0	2.0201	1.00
	2.5	2.03	1.00
	3.0	2.04	1.0345
	3.5	2.0462	1.0971
	4.0	2.05	1.1489
	4.5	2.0539	1.19
	5.0	2.0566	1.2343
AS6461	1.0	2.185	1.00
	1.5	2.4779	1.00
	2.0	3.528	1.00
	2.5	3.85	1.00
	3.0	3.88	1.00
	3.5	3.898	1.00
	4.0	3.906	1.00
	4.5	3.9098	1.00
	5.0	3.912	1.00
nsf	1.0	1.600	1.00
	1.5	2.52	1.4202
	2.0	3.0025	1.6069
	2.5	3.202	1.6804
	3.0	3.33	1.7349
	3.5	3.43	1.7711
	4.0	3.5012	1.7945
	4.5	3.5567	1.8016
	5.0	3.6010	1.8040
AS4755	1.0	2.1098	1.00
	1.5	2.1176	1.00
	2.0	2.121	1.00
	2.5	2.123	1.00
	3.0	2.1243	1.00
	3.5	2.1253	1.00
	4.0	2.1260	1.00
	4.5	2.1266	1.00
	5.0	2.1270	1.00

Table 5.3: congestion ratio for Robust and Optimal routing solutions



Figure 5.4: Congestion ratio for ISP topology AS4755

It is apparent from the figures 5.1-5.4 that in all cases the robust routing has a congestion ratio that is worse than the optimal routing congestion ratio,computed for the topologies. With low degree of uncertainty, robust routing performs sensibly with the reasonable performance loss. On the contrary, optimal routing has a congestion ratio of 1 in most cases.

We have noticed in the Figure 5.1 that with the lower degree of uncertainty the congestion ratio for the robust routing is 1.96 and for the optimal routing is 1.00 for AS1221 network. This shows a performance loss of 96%. Whereas, the performance loss declines to 60% on varying the value of p to 5.0. In other cases, we observed that for AS6461 network the robust routing attains the ratio that approaches to 4.0 while optimal routing stabilizes to the CR value at 1.0.

Similarly, for nsf network, robust routing obtains a good performance at lower degree of uncertainty with performance loss up to 60%. The congestion ratio for the robust routing is 1.6 when p is 1.0 compared to value 1.0 for optimal routing. The robust routing performance declines steadily on varying the parameter p. It can be seen from the Figure 5.3, that congestion ratio rises to 3.6 for the robust routing compared to 1.8 for optimal one.

5.7 Discussion

As we described earlier, sudden variation in demand can have a significant impact on the core network performance.Instabilities that reflect sudden and voluminous demand variation may lead to increased network congestion, packet loss or additional CPU overheads on the routers. We defined a robust routing problem for discussing routing information and suggested a routing solution that may account for some anomalous behaviors.

Using the concept of *oblivious routing*, we first determined the minimization of maximum link utilization for the network topology and then used the robust routing algorithm to design for dynamic demand variations. Our routing procedure is based on evenly redistributing the network load between pairs of nodes and the resulting performance is measured as congestion ratio.

To validate our results, we used mathematical modelling tool AMPL / CPLEX 10.2. To tackle with the sudden variations which include DDoS or worms, we modelled the demand polytope around the base traffic demand to capture the variations in equation number 5.20 and 5.21. We performed several experiments using an uncertainty parameter and solved the mathematical program for optimal and robust solution.

We conducted experiments using four topologies. Through our computa-

tional experiments, we found that the objective that is congestion ratio remains nearly 2.0 for two of the four topologies when the uncertainty parameter is varied in steps from 1.0 to 5.0. Higher value of p implies higher demand variations.

We may not be able to make conclusive judgement on the performance of the proposed robust routing based on the sample of ISP topologies. However, we gain high confidence that the proposed path selection method can have close approximation to [5].

5.8 Summary

We presented robust routing to tackle with sudden and voluminous demand variations. Some of the robust routing algorithms were implemented on network models considering cost as a measure of network performance. From our studies, we found that with efficient use of interdomain traffic resources and incremental cost model, one can expect a robust network design to respond to the dynamic demand variations without sacrifying network performance objectives. A similar approach based on the uncertain traffic demand has been proposed in [5, 12]. In contrast to the previous work, the proposed robust routing includes the presence of DDoS attacks which is relatively a new idea.

It is quite challenging to test the robustness in terms of computational complexity if optimization problem is constrained by larger uncertain demand set. It always remains an open question that how to choose a proper demand polytope or subset of larger demand set to produce an efficient and feasible robust optimization. Possible future work may be to investigate the trade off between cost versus optimization of maximum link utilization.

We believe, further studies are crucial for gaining insight into the robust routing behavior and the network performance so that a rational growth of the Internet can be sustained. It will be interesting to extend the proposed robust routing as a cooperative routing platform by integrating the overlay routing. Chapter 6

Conclusion and Future Directions

We summarize the main contributions of this dissertation and present some interesting future directions in the area of research.

6.1 Conclusion

In the past decade, rapid developments in the technologies and increase in the Internet applications have been driving an ever-increasing request for bandwidth at the edge of the network, accompanied with the increase in the need for quality of service and robustness to the traffic variations. Some of the motivations behind the introduction of oblivious routing and subsequent related work, is the desire to design a routing framework that can combine the best of two worlds: provide robustness to the changing demand and having the benefits of partitioning the demand uncertainty into smaller polytopes for periodic demand surges.

In this thesis, we study the problem of routing with robustness to the dynamic traffic demands from various perspectives.Our main contributions are as follows:

- * Given an intradomain network topology, we presented a systematic way to allocate the network traffic on multiple paths in a network. It has been observed that Internet routing when subjected to the traffic demand variations, may apparently lead to congestion in a network. We formulated the robust routing problem with an objective of minimizing maximum link utilization(MLU). After that, we presented a simple routing algorithm called robust routing technique(RRT) to select multiple routing paths with balancing load on the links of a network. We use three routing methods in the implementation: Equal Cost Multiple Path(ECMP), Widest Shortest Path(WSP), and K-Shortest Path(KSET) for the allocation of traffic to compute the maximum link utilization motivated by routing robustness with variability in the traffic demand to manage the multi-commodity flows between origin-destination routers in a core network. The routing methods we propose are useful in setting network performance in different scenarios when there is a traffic demand variation. In order to show the effectiveness of our routing solution we used real network data and synthetic data.
- * In addition, we study the routing performance of RRT on different scenarios using synthetic topologies. This includes the transit-stub graphs to capture the interdomain routing. We presented a simple routing framework to minimize MLU over transit-stub graphs.Simulation results show the effectiveness of our algorithm and illustrate that transit-stub graphs can utilize network efficiently on performing our

algorithm.

* Furthermore, we study the robust optimization problem from the perspective of intradomain traffic engineering. We conducted a systematic study on traffic engineering using OSPF style routing and oblivious routing. We formulated the robust routing problem under dynamic traffic demand which is modelled to capture the dynamic variations e.g. DDoS or worms. We present an efficient algorithm to compute congestion ratio for given routing paths. We demonstrated the performance of algorithm on real network topologies. Our routing framework is flexible to integrate the MPLS style routing.

6.2 Future directions

Robust traffic engineering is still in the evolving phase before being deployed as next generation Internet services. Future work will be in many possible directions. One of the stumbling blocks for the future development is a co-operative network architecture between robust traffic engineering for the core network and overlay routing. In our previous studies we focused mainly on the robust routing with optimization for the core network. Our routing methods do not consider information of the overlay routing. Internet is a heterogeneous and competitive in nature, hence integration of robust TE with overlay routing is a challenging task for the next generation Internet.

We study robust routing on a data plane. On the contrary, overlay routing decisions are mainly based on the selfish routing applying individual probe for the information they need. However, we believe that by incorporating cooperative decision making platform between core network routers and overlay probing, overall network performance can be improved. In our study, we gave a traffic engineering solution for an objective of minimizing maximum link utilization. We proposed, how to minimize the MLU among multiple competing routing paths. One of the interesting problems is to explore the robust traffic engineering in the core network constrained by the overlay path selection at all ISPs between end hosts. However this requires a detailed study on the network design aspects of data plane(for the robustness in core network) and application plane(for overlay routing).

Another interesting direction is to consider the ISP's economical motivation to support our robust routing optimization. Robust routing optimization solutions may prove expensive if they are applied within ISP core. This may conflict with the neighboring ISP in terms of economics of operation. Therefore it will be useful to establish a trade off between cost versus robust routing optimization.

Appendix

A.1 Code of the robust optimization problem in chapter 5 #Robust routing problem with demand uncertainty #This is RobustRouting.run file: running in batch mode

```
model RobustRouting.mod;
data 6461.dat;
option solver cplexamp;
#### reduce graph
for {i in nodes} {
let repel [i] := sum {j in nodes: (i,j) in arcs} trace [i,j];
let attract [i] := sum {j in nodes: (j,i) in arcs} trace [j,i];
}
param flag default 1;
repeat while flag = 1 {
let flag := 0;
for {i in V: card ({j in V: (i,j) in E or (j,i) in E}) < 2} {</pre>
```

```
let flag := 1;
for \{j \text{ in } V: (i,j) \text{ in } E \text{ or } (j,i) \text{ in } E\}
let repel [j] := repel [j] + repel [i];
let attract [j] := attract [j] + attract [i];
}
let V := V \operatorname{diff} \{i\};
}
}
### Calculate the shortest path according to inverse capacity rule i.e.
1/c
param sumr default sum {j in V} repel [j];
param suma default sum {j in V} attract [j];
for \{i \text{ in } V\} let repel [i] := \text{repel } [i] / \text{sumr};
for \{i \text{ in } V\} let attract [i] := \text{attract } [i] / \text{suma};
\#\#\# shortest path problem definition
var path {A, odpairs} >= 0 <= 1;
param s in nodes;
param t in nodes;
minimize plen: sum \{(h,k) \text{ in } E\} (path [h,k,s,t] + \text{path } [k,h,s,t]) / c [h,k];
flow_cons \{h \text{ in } V\}:
sum \{(h,k) \text{ in } A\} (path [h,k,s,t] - path [k,h,s,t])
= if (h = s) then 1
else (if (h = t) then - 1
else 0;
```

```
problem shortest_path: path, plen, flow_cons;
printf "======== Computing shortest paths\n";
for \{(i,j) \text{ in odpairs}\}
let s := i;
let t := j;
problem shortest_path;
option cplex_options 'timing=0 lpdisplay=1';
solve shortest_path > outputshortestpath6461.txt;
for {(h,k) in A: path [h,k,i,j] > 1e-10} {
let ff [h,k,i,j] := path [h,k,i,j];
}
}
\#\#\# compute the initial demand
var g {A, odpairs} \geq = 0;
var betamax >= 0;
maximize beta0: betamax;
capacity \{(h,k) \text{ in } E\}: sum \{(i,j) \text{ in odpairs}\} (g [h,k,i,j] + g [k,h,i,j]) <=
c [h,k];
beta_flow_conservation {h in V, (i,j) in odpairs}:
sum {(h,k) in A} (g [h,k,i,j] - g [k,h,i,j])
= if (h = i) then ( betamax * repel [i] * attract [j])
else (if (h = j) then (- betamax * repel [i] * attract [j])
else 0;
```

problem init_dbar; option cplex_options 'baropt bardisplay=1 timing=1 lpdisplay=1';

problem init_dbar; option cplex_options 'baropt bardisplay=1 timing=1 lpdisplay=1';

solve init_dbar;

let beta := betamax;

param optimal_cr;

let worst := cr;

problem robust: r, f,

pi, sigma, eta, lambda, mu, chi,

opr,

flow_conservation, perf_ratio, g_dual, D_dual, omega_dual;

problem robust; option cplex_options 'baropt bardisplay=1 timing=1 lpdisplay=1';

problem robust; option cplex_options 'baropt bardisplay=1 timing=1 lpdisplay=1';

solve robust > outputrobust6461.txt;

let $optimal_cr := cr;$

A.2 RRT Algorithm rrt.c

```
* Copyright (c) 2008
*
* Author: Duc Quang Bui (duc.bui@student.rmit.edu.au), Jul. 2008
* Himanshu Agrawal (himanshu.agrawal@rmit.edu.au)
*
#include "rrt.h"
//The main function for the Robust Routing Test Algorithm///
long RRT_robust_routing_test(char *rt_mode)
{
TT *tt;
Path *first_P, *P, *X, *tmp_P;
Graph *gg;
Vertex *vi, *ve, *tmp_ve;
long org_weight=0, cur_weight, k, i;
//\mathrm{RRT} running under Weighted Shortest Path Mode
if (strcmp (rt_mode, "wsp") == 0) { //WSP Mode
```

```
printf("Robust Routing Test is using simple Weighted Shortest Path
Scheme : RRT-WSP\n");
```

tt = firstTT;

```
while (tt) {
```

 $vi = tt - v_i;$

 $ve = tt -> eg_node -> vert;$

```
//printf("Traffic Trace No. %d from V(%s) -> V(%s)\n", tt->id, tt-
```

```
>v_in->name, tt->eg_node->vert->name);
```

```
org_weight = wsp(vi, ve, test_bed, NULL);
```

```
//print_dijkstra_result(ve);
```

```
RRT_Allocate(tt, test_bed, 1);
```

```
tt = tt - next;
```

```
}
```

}

 $//GRP_print(test_bed);$

/*********************************/

// RRT running under ECMP Mode

```
/****************************/
```

else if (strcmp (rt_mode, "ecmp") == 0) { //ECMP Mode using K-SP Algorithm

//GRP_print(test_bed);

printf("Robust Routing Test is using ECMP Scheme : RRT-ECMPn"); tt = firstTT;

while (tt) {

 $vi = tt - v_i;$ $ve = tt -> eg_node -> vert;$ //printf("Traffic Trace No. %d from V(%s) -> V(%s)\n", tt->id, tt->v_in->name, tt->eg_node->vert->name); $first_P = (Path^*) ecmp(vi, ve, test_bed);$ $P = first_P;$ i = 0;//Allocate traffic to all traffic trunks while (P) { X = P - next;//printf("%d/%d/%d - ", i, P->id, P->weight); $\mathbf{P}=\mathbf{X};$ i++; } $P = first_P;$ while (P) { X = P->next; //printf("%d/%d/%d - ", i, P->id, P->weight); //tt->g = P->graph; $tt \rightarrow v_eg = P \rightarrow v_eg;$ $tt \rightarrow v_i = P \rightarrow v_i;$ $tt \rightarrow eg_node \rightarrow vert = P \rightarrow v_eg;$ RRT_Allocate(tt, test_bed, i); if (P->id == 1) {// we keep the graph test_bed

```
printf("%d/%d/%d - ", i, P->id, P->weight);
free(P);
}
else \{
printf("%d/%d/%d - ", i, P->id, P->weight);
GRP_free_graph_ext(P->graph);
gb_recycle(P->graph);
free(P);
}
\mathbf{P} = \mathbf{X};
}
tt = tt - next;
printf("\n---
                                 --\n");
}
}
//RRT using k-set-INVCAP traffic allocation under ECMP Mode
else if (strcmp (rt_mode, "kset_invcap") == 0) { //k-set-INVCAP Mode
printf("Robust Routing Test is using k-set-INVCAP Scheme : RRT-k-
set-INVCAPn");
for (k = 0; k < 10; k++) {
GRP_update_weight(test_bed);
tt = firstTT;
```

```
while (tt) {
```

```
vi = tt - v_i;
```

 $ve = tt -> eg_node -> vert;$

//printf("Traffic Trace No. %d from V(%s) -> V(%s)\n", tt->id, tt->v_in->name, tt->eg_node->vert->name);

```
//org_weight = dijkstra(vi, ve, test_bed, NULL);
```

```
org_weight = wsp(vi, ve, test_bed, NULL);
```

```
//print_dijkstra_result(ve);
```

```
RRT_Allocate(tt, test_bed, 2);
```

```
RRT_Update_Bandwidth(tt, tt->ingress_bandwidth/2);
```

```
tt = tt - next;
```

```
}
```

```
}
```

```
}
```

else

{

```
error_exit("[Routing protocol is not a correct mode");
```

```
GRP_print(test_bed);
```

 $//{\rm This}$ function reads traffic matrix in the input file

```
int RRT_read_traffic_matrix(char *f)
```

```
FILE *fp; /* the traffic trace file */
```

double band_width;

long id, prev_id=-1, j=0, k;

char *i, *e;

 $/\ast$ we have to convert ing and eg into Vertices $\ast/$

Vertex *vi,*ve;

/* counts the lines and chars in each line of the Traffic Matrix File */

 $long ln=0, ln_chars=0;$

 $/\ast$ where we read each line and each traffic trace value $\ast/$

char *buf, *trace;

TT *tt; /* pointer the newly added (everytime) tt */

unsigned char first_line = 1;

char ch;

if ((fp=fopen(f,"r")) == (FILE *)NULL)

error_exit("Cannot open traffic traces file!!!");

while $(\text{fscanf}(\text{fp}, \ensuremath{"\%c"}, \&ch) == 1)$ {

if (ch != COMMENT) {

while $((\text{fscanf}(\text{fp}, \ensuremath{''\%c''}, \&\text{ch}) == 1) \&\& (\text{ch} \mathrel{!= '\n'}))$

ln_chars++;

/* reserve some more spaces for characters at the end of lines */ ln_chars += 20; printf("////%d////\n", ln_chars); break;

}

while $((\text{fscanf}(\text{fp}, "\%c", \&ch) == 1) \&\& (ch != '\n'));$

```
i = (char *)malloc(sizeof(char)*(find_digits(test_bed->m)+1));
e = (char *)malloc(sizeof(char)*(find_digits(test_bed->m)+1));
if ((buf = (char^*)malloc(sizeof(char)^*ln_chars)) = (char^*)NULL)
error_exit("Cannot allocate space for buf");
//Count the number of traffic entries, i.e. number of traffic trace lines
fseek(fp, 0, 0);
while (fgets(buf, ln_chars, fp) != (char *)NULL ) {
if (buf[0] != COMMENT)
ln++;
}
//printf("||||||%d|||||\\n", ln);
fseek(fp, 0, 0);
while (fgets(buf, ln_chars, fp) != (char *)NULL ) {
if (buf[0] != COMMENT) {
for (trace = (char^*)strtok(buf,""), k = 0; trace != NULL; trace =
(char*)strtok
(NULL, " "), k++) {
//printf("%s ", ch);
if (k < ln) {
//for (k = 0; k < ln; k++) \{
band_width = (double)atof(trace);
sprintf(i, "\%d \setminus 0", j);
sprintf(e, "\%d \setminus 0", k);
if (j != k) {
```

}

```
91
```

```
if( (vi= hash_lookup(i,test_bed)) == (Vertex *)NULL )
fprintf(stderr,"Warning!!Not valid ingress(%s:%ld)\n",f,j);
else
if( (ve= hash_lookup(e,test_bed)) == (Vertex *)NULL )
fprintf(stderr,"Warning!!Not valid egress(%s:%ld)\n",f,j);
else /* both valid vertices */
{
tt = RRT_create(j,vi,ve,band_width);
}
}
}
}
j++;
}
\} /* while */
free(buf);
free(i);
free(e);
}
//This function reads weights of links in the input topology file
void RRT_read_weight(char *f)
{
```

/* the traffic trace file */

FILE *fp;

double weight;

long j=0, k, ii;

char *i, *e;

/* we have to convert ing and eg into Vertices */

Vertex *vi,*ve;

Arc *a;

/* counts the lines and chars in each line of the Traffic Matrix File */

 $long ln=0, ln_chars=0;$

/* where we read each line and each traffic trace value */

char *buf, *trace;

char ch;

printf("RRT_read_weight(%s)\n", f);

if ((fp=fopen(f,"r")) == (FILE *)NULL)

error_exit("Cannot open weight file!!!");

while $(fscanf(fp, "\%s", \&ch) == 1) \{$

while $((\text{fscanf}(\text{fp}, "\%c", \&ch) == 1) \&\& (ch != '\n'))$

 $ln_chars++;$

/* reserve some more spaces for characters at the end of lines*/ ln_chars += 20;

 $\label{eq:printf} //printf("////%d////\n",\ln_chars);$

break;

}

 $i = (char *)malloc(sizeof(char)*(find_digits(test_bed->m)+1));$ $e = (char *)malloc(sizeof(char)*(find_digits(test_bed > m)+1));$ if $((buf = (char^*)malloc(sizeof(char)^*ln_chars)) = (char^*)NULL)$ error_exit("Cannot allocate space for buf"); //Count the number of traffic entries, i.e. number of traffic trace lines fseek(fp, 0, 0);while (fgets(buf, ln_chars, fp) != (char *)NULL) ln++;//printf("||||||%d||||||\n", ln); fseek(fp, 0, 0);while (fgets(buf, ln_chars, fp) != (char *)NULL) { for (trace = (char*)strtok(buf," "), k = 0; trace != NULL; trace = (char*)strtok (NULL, " "), k++) { //printf("%s ", trace); if (k < ln) { weight = (double)atof(trace);if (weight > -1) { $sprintf(i, "\%d \setminus 0", j);$ $sprintf(e, "\%d \setminus 0", k);$ if (j != k) { if((vi= hash_lookup(i,test_bed)) == (Vertex *)NULL) fprintf(stderr,"Warning!!Not valid ingress(%s:%ld)\n",f,ln); else

if((ve= hash_lookup(e,test_bed)) == (Vertex *)NULL)

```
fprintf(stderr,"Warning!!Not valid egress(%s:%ld)\n",f,ln);
else /* both valid vertices */
{
for (a = vi-arcs; a\&\&(a-tip!=ve); a=a-ave);
if (!a)
fprintf(stderr,"Warning!!Not valid arc (%s->%s %s:%ld)\n",i, e, f,j);
a > len = weight;
}
}
}
}
}
j++;
} /* while */
free(buf); free(i); free(e);
}
// This section of code creates the traffic
TT* RRT_create(long id, Vertex *vi, Vertex *ve, double bw)
{
TT *new_tt;
NodeExt *eg_vj;
if ( (\text{new}_{tt} = (\text{TT }^*)\text{malloc}(\text{sizeof}(\text{TT}))) = (\text{TT }^*)\text{NULL} )
```
```
error_exit("TT_create(1)");
new_tt - next = firstTT;
firstTT = new_tt;
new_tt->v_in = vi; /* This TT is identified by its Igress Node */
new_tt->ingress_bandwidth = bw;
new_tt->id = id;
//\text{new}_\text{tt->g} = \text{NULL};
new_tt \rightarrow v_eg = NULL;
// add the first egress to the new tt
if ( (eg_vj=(NodeExt^*)malloc(sizeof(NodeExt))) == (NodeExt^*)NULL
)
error_exit("TT_create(2)");
new_tt->eg_node=(NodeExt*)eg_vj; /* tti carries the address */
/* of the first egress node*/
eg_vj->vert = ve; /* Egress Node */
eg_vj->next = NULL; /*Point to Next Egress Node - It can be deleted
*/
eg_vj->eg_path = NULL; /*Point to Chain of Allocated Struct initialised
*/
return new_tt;
}
```

```
//This function allocates traffic to all traffic trunks
void RRT_Allocate(TT* tti, Graph *g, long k)
{
Vertex*t,*p,*q, *vv, *vi, *vj, *vii, *vj;
Arc *a;
t = NULL, vi = NULL;
p=vv=tti->eg_node->vert;
if(!p->back_link){
printf("(allocate)Sorry, %s is unreachable.\n",p->name);
return;
}
do\{ q= p-back_link;
p->back_link= t;
t = p;
p = q;
}while(t!=p);
do{
vj=t;
if(vi==NULL)
vi=vj;
if(vi!=vj){
vii = hash_lookup(vi > name, g);
vjj = hash_lookup(vj->name, g);
```

```
for (a = vii-arcs; a\&\&(a-tip != vjj); a=a-atic);
if(!a)
error_exit("RRT_Allocate()");
((ArcExt^*)a->a.A)->cur_load += (tti->ingress_bandwidth/k);
// RRT_Update_Allocation(vi, vj, tti, tti->eg_node, a, tti->ingress_bandwidth);
vi=vj;
}
t= t->back_link;
}while(t);
//printf("\n");
t = p;
do{
q = t - back_link;
t->back_link= p;
p = t;
t = q;
}while(p!=vv);
}
//This function updates the bandwidth field in the traffic trunk structure
RRT_Update_Bandwidth(TT* tti, long bw)
{
tti->ingress\_bandwidth = bw;
```

```
}
//Recalculating the traffic allocation for each link related to the traffic
trunk
void RRT_Update_Allocation(Vertex* vii, Vertex *vjj, TT* tti, NodeExt
*eg_vj, Arc *arc, double bw_to_be_allocated){
static Vertex *Previous_vjj = 0;
static TT *Previous_tti = 0;
static allocated_path *last_all = 0;
}
allocated_path *current_al_p;
current_al_p=(allocated_path*) malloc(sizeof(allocated_path));
if (current_al_p = NULL){
perror("TT_Update_Allocation():There is no memory available");
perror("Sorry, This is a fatal error");
exit(??);
}
if ((tti==Previous_tti) && (Previous_vjj==vii)){
last_all->next=current_al_p;
} else{
eg_vj->eg_path=current_al_p;
}
current_al_p->next=0;
```

```
99
```

```
current_al_p->bandwidth_allocated=bw_to_be_allocated;
current_al_p->vi=vii;
current_al_p->vj=vjj;
current_al_p->arc=arc;
Previous_vjj=vjj;
Previous_tti=tti;
last_all=current_al_p;
//printf("Allocate (%s)-%4.4fMb->(%s)\n", vii->name, bw_to_be_allocated,
vjj->name);
}
//This function writes to the output file readable for iLOG software
void RRT_output_iLOG(FILE *f)
{
TT *tti, *tmp_tt;
Vertex *u, *v;
Arc *a, *b;
int i, j, k, l=1, id, prev_id;
long Load=0, Sum_ai=0, Sum_bi=0, Av_Over=0;
double ai, bi=0, bii=0, di, MLU=0;
fprintf(f, "param N_NODES := \%d n", test_bed->n);
fprintf(f, "set EDGES :=");
j = 0;
```

```
for (v = test_bed->vertices, i=0; i < test_bed->n; i++, v++) {
for (a = v - arcs; a; a = a - next)
u = a -> tip;
if (((ArcExt*)a->a.A)->mrk == NOLOCK) {
((ArcExt^*)a->a.A)->mrk = LOCK;
for (b = u - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b) \&\& (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v); b = b - arcs; (b - tip != v
if (!b)
error_exit("RRT_output_iLOG()");
((ArcExt^*)b \rightarrow a.A) \rightarrow mrk = LOCK;
fprintf(f, "(\%s,\%s)", v->name, u->name);
j++;
if (j == 10) {
fprintf(f, "\n");
j = 0;
}
 }
}
 }
fprintf(f, "; n");
fprintf(f, "param distance:=\n");
for (v = test_bed->vertices, i=0; i < test_bed->n; i++, v++) {
for (a = v - arcs; a; a = a - arcs)
u = a -> tip;
if (((ArcExt^*)a->a.A)->mrk == LOCK) {
```

```
((ArcExt^*)a->a.A)->mrk = NOLOCK;
for (b = u - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b) \&\& (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = b - xcs; (b - tip != v); b = xcs; (b - tip != v); (b - tip != xcs; (b - tip != xcs; (b - tip != v); b = xcs; (b - tip != xcs
if (!b)
error_exit("RRT_output_iLOG()");
((ArcExt^*)b > a.A) > mrk = NOLOCK;
fprintf(f, "%s %s %6.2f\n", v->name, u->name, ((ArcExt*)a->a.A)-
>cur_load);
}
}
 }
fprintf(f, "; n");
}
 //{\rm This} function frees all dynamic memory allocation
void RRT_free_TT(TT *tti)
{
TT *tmp_tt;
while (tti) {
tmp_tt = tti;
tti = tti - next;
free(tmp_tt->eg_node);
free(tmp_tt);
}
```

}

A.3 Abbreviation used in the thesis

We used following abbreviations throughout our thesis:

Abbreviation	Description
ISP	Internet Service Provider
TE	Traffic Engineering
QoS	Quality of Service
P2P	Point to Point
OSPF	Open Shortest Path First
MPLS	Multi Protocol Label Switching
AS	Autonomous System
TM	Traffic Matrix
MLU	Maximum Link Utilization
CR	Congestion Ratio

Bibliography

- R. K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows*. Prentice - Hall, 1993.
- [2] A. Altin, E. Amaldi, P. Belloti, and M. C. Pinar. Provisioning virtual private networks under traffic uncertainty. *Networks*, 49(1):100–115, 2007.
- [3] D. Applegate, L. Breslau, and E. Cohen. Coping with network failures: routing strategies for optimal demand oblivious restoration. Proceedings of the joint international conference on Measurement and modelling of computer systems, pages 270–281, 2004.
- [4] D. Applegate and E. Cohen. Making Intradomain Routing Robust to Changing and Uncertain Traffic Demand: Understanding Fundamental Tradeoffs. In Proc. SIGCOMM, pages 313–324, 2003.
- [5] D. Applegate and E. Cohen. Making Routing Robust to Changing Traffic Demands: Algorithms and Evaluation. *IEEE/ACM TRANS-ACTIONS ON NETWORKING*, 14(6):1193–1206, December 2006.
- [6] G. Ash. Dynamic Routing in Telecommunication Networks. McGraw
 Hill, 1998.

- [7] G. R. Ash, F. Chang, and D. Medhi. Robust traffic design for dynamic routing networks. In Proc. IEEE INFOCOM'91, pages 508– 514, April 1991.
- [8] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and Principals of Internet Traffic Engineering. *RFC3272*, May 2002.
- [9] D. Awduche and J. Malcolm. MPLS and Traffic Engineering in IP networks. *IEEE Communication Magazene*, 37(12):42–47, December 1999.
- [10] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements of traffic engineering over MPLS. *RFC2702*, September 1999.
- [11] Y. Azar, E. Cohen, A. Flat, H. Kalplan, and H. Racke. Optimal Oblivious Routing in Polynomial Time. In Proc. 35th ACM Symposium on the Theory of Computing, pages 237–245, 2003.
- [12] P. Belloti and M.C. Pinar. Optimal Oblivious Routing under linear and ellipsoidal uncertainty. *Optimization and Engineering*, 2005.
- [13] P. Belloti and M.C. Pinar. Optimal Oblivious Routing under statistical uncertainty. Optimization and Engineering, 9(3):257–271, 2008.
- [14] Ben-Ameur. Between Fully Dynamic Routing and Robust Stable Routing. In Proc. DRCN, 2007.
- [15] W. Ben-Ameur and H. Kerivin. Routing of uncertain traffic demands. Proc. Informs, 2001.
- [16] W. Ben-Ameur and H. Kerivin. Routing of Uncertain Demands. Optimization and Engineering, 6(3):283–313, September 2005.

- [17] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. Operation Research Letters, 25(1):1–13, 1999.
- [18] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.
- [19] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, Ser. B 98:43–71, 2003.
- [20] Caida. [online]. Available: http://www.caida.org/data/realtime/.
- [21] P. Casas and S. Vaton. An Adaptive Multi Temporal approach for Robust Routing. In Proc. Euro-FGI Workshop on QoS and Traffic Control, 2007.
- [22] C. Chekuri, F. B. Shepherd, G. Oriolo, and M. G. Scutella. Hardness of robust network design. *Networks*, 50(1):50–54, April 2007.
- [23] E. W. Dijkstra. A note on two problems in connections with graphs. Numerische Mathematik, 1:269–271, 1959.
- [24] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, and J.E. van der Merwe. A flexible model for resource management in virtual private network. *Proceedings of ACM SIGCOMM Computer Communication Review*, 29:95–109, April 1999.
- [25] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS Adaptive Traffic Engineering. In Proceedings of INFOCOMM, pages 1300– 1309, 2001.
- [26] M. Ericsson, M. G. C. Resende, and P. M. Pardalos. A Genetic Algorithm for the Weight Setting Problem in OSPF routing. *Journal* of Combinatorial Optimization, 6(3):299–333, September 2002.
- [27] N. Feamster, J. Borkenhagan, and J. Rexford. Guidelines for Interdomain Traffic Engineering. ACM SIGCOMM Computer Commu-

nication Review, 33(5):19–30, October 2003.

- [28] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: methodology and experience. *IEEE/ACM Transactions of Networking*, 9(3):265–280, 2001.
- [29] S. Floyd. Congestion Control Principles. *IETF RFC 2914*, September 2000.
- [30] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF weights. In Proc. IEEE INFOCOMM, pages 519–528, March 2000.
- [31] B. Fortz and M. Thorup. Optmizing OSPF/IS-IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, May 2002.
- [32] L. Gao and J. Rexford. Stable Internet Routing without global coordination. *IEET/ACM Transactions on Networking*, 9(6):681–692, December 2001.
- [33] I. Gojmerac, T. Ziegler, F. Ricciato, and P. Reichi. Adaptive Multipath Routing for Dynamic Traffic Engineering. In Proc. IEEE GLOBECOM 2003, 2003.
- [34] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford,
 G. Xie, H. Yang, J. Zhan, and H. Zhang. A clean slate 4D approach to network control and management. SIGCOMM Comput. Commun. Rev., 35(5):41–54, 2005.
- [35] GTITM. Gt-itm: Georgia tech internetwork topolgy models. Available: http://www.cc.gatech.edu/gt-itm/gt-itm.tar.gz.

- [36] J. He, M. Chiang, J. Rexford, E. E. Router, and E. Link. DATE:Distributed Adaptive Traffic Engineering. *Poster session at INFOCOM*, 2005.
- [37] K. Holmberg and D. Yuan. Optimization of internet protocol network design and routing. *Networks*, 43(1):39–53, 2004.
- [38] C. Huitema. Routing in the Internet. Prentice Hall, 2000.
- [39] ILOG. ILOG CPLEX Large Scale mathematical programming software. Available: http://www.ilog.com/.
- [40] J. Jaffe and F. Moss. A responsive distributed routing algorithm for computer networks. *IEEE Trans. on Communications*, 30:1758– 1762, 1982.
- [41] M. Johansson and A. Gunnar. Data-driven Traffic Engineering: techniques, experiences and challenges. In Proc. BROADNETS, 2006.
- [42] I. Juva. Robust Load Balanacing. In Proc. GLOBECOM, 2007.
- [43] S. Kandula, D. Katabhi, B. Davie, and A. Charny. Walking the Tightrope: Responsive yet Stable Traffic Engineering. In Proceedings of SIGCOMM, pages 253–264, 2005.
- [44] S. Khullar and C. Chekuri. Routing and Network Design with Robustness to Changing or Uncertain Traffic Demands. ACM SIGACT News, 38(3):106–128, 2007.
- [45] D. E. Knuth. The Stanford Graphbase: A Plateform for Combinatorial Computing. 1993.
- [46] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Trans. on Networking*, 6:515–528, 1998.

- [47] Y. Li, B. Bai, J. Harms, and R. Holte. Multipath Oblivious Routing for Traffic Engineering. *IEEE/ACM Transactions on Networking*, 2007.
- [48] D. Medhi and K. Ramasamy. Network Routing : Algorithms, Protocols, and Architectures. Morgan Kaufman Publishers, 2007.
- [49] A. Medina, N. Taft, K. Salamatian, S. Bhattacharya, and C. Diot. Traffic Matrix Estimation: Existing Techniques and New Directions. In Proc. SIGCOMM, 2002.
- [50] Jelena Mirkovic, Sven Dietrich, David Dittrich, and Peter Reiher.
 Internet Denial of Service Attack and Defence Mechanisms. Prentice
 Hall, 2005.
- [51] J. Moy. OSPF: Anatomy of an Internet Routing Protocol. Addition-Wesley, 1998.
- [52] J. Moy. OSPF version 2. *IETF RFC 2328*, April 1998.
- [53] H. Nagesh, V. Poosala, V. Kumar, P. Winzer, and M. Zirngibl. Load balanced architecture for dynamic traffic. Optical Fiber Communication Conf. (OFC), 2004.
- [54] J. Nagle. Congestion Control in IP/TCP. IETF RFC 896, January 1984.
- [55] ns 2. homepage. Available: http://www.isi.edu/nsnam/ns/.
- [56] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP performance metrics. *Request for Comments 2330*, 1998.
- [57] G. Petrou, C. Lemarechal, and A. Ouorou. Robust network design in telecommunications. Proc. International Network Optimization Conference, 2005.

- [58] M. Pioro and D. Medhi. Routing Flow and Capacity Design in Communication and Computer Networks. Morgan Kaufman Publishers, 2004.
- [59] H. Racke. Minimizing Congestion in General Networks. In Proc. IEEE FOCS, pages 43–52, 2002.
- [60] K. G. Ramakrishnan and M. A. Rodrigues. Optimal routing in shortest-path data networks. *Bell Labs Technical Journal*, 6(1):117– 138, 2001.
- [61] G. Retvari and T. Cinkler. Practical OSPF Traffic Engineering. IEEE Communications Letters, 8(11):689–691, November 2004.
- [62] J. Rexford. Route Optimization in IP networks. In Handbook of Optimization in Telecommunications, 2006.
- [63] M. Roughan, M. Thorup, and Y. Zhang. Traffic Engineering with Estimated Traffic Matrix. In the 3rd ACM SIGCOMM Conference on Internet Measurement, pages 248–258, 2003.
- [64] F. B. Shepherd and P. Winzer. Selective Randomized Load Balancing and Mesh Networks with Changing Demands. *Journal of Optical Networking*, 5(5):320–339, 2006.
- [65] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In Proc. ACM SIGCOMM, 2002.
- [66] A. Sridharan, R. Guerin, and C. Diot. Achieving near-optimal traffic engineering solutions for current OSPF-/IS-IS networks. *IEET/ACM Transactions on Networking*, 13(2):234–247, April 2005.
- [67] R. Teixeira, A. Shaikh, and T. Griffin and. Dynamics of hot-potato routing in IP networks. ACM SIGMETRICS Performance Evalua-

tion Review, 32(1):307-319, 2004.

- [68] K. Thompson, G. J. Miller, and R. Wilder. Wide area Internet traffic patterns and characteristics. *IEEE Network Mag.*, pages 10–23, 1997.
- [69] A. Tizghadam and A. Leon-Garcia. On Robust Traffic Engineering in Core Networks. In Proc. IEEE GLOBCOMM, 2008.
- [70] Y. Vardi. Network Tomography:estimating source-destination traffic intencities from link data. Journal of American Statistical Association, 91, 1996.
- [71] H. Wang, H. Xie, L. Qui, Y. R. Zhang, and A. Greenberg. COPE: Traffic Engineering in Dynamic Networks. In Proceeding of SIG-COMM, 2006.
- [72] J. Wang, Y. Yang, and K Nahrastedt. Edge-based traffic engineering for OSPF networks. *Computer Networks*, 48(4):605–625, July 2005.
- [73] X. Xiao, A. Hannan, B. Bailey, and L. Ni. Traffic Engineering with MPLS in the Internet. *IEEE Network Mag.*, pages 28–33, 2000.
- [74] Y. R. Yang, H. Xie, H. Wang, A. Silberschatz, A. Krishnamurthy, Y. Liu, and L. E. Li. On route selection for interdomain traffic engineering. *Network, IEEE*, 19(6):20–27, Nov.-Dec. 2005.
- [75] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, and D. Towsley. On optimal routing with multiple traffic matrices. *In Proc. IEEE INFOCOMM*, March 2005.