MULTIPATH ROUTING AND LOAD SHARING
USING GAME THEORY

by

Kunpeng Liu
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Electrical and Computer Engineering

Committee:

_____ Dr. Bijan Jabbari, Dissertation Director

_____ Dr. Shih-Chun Chang, Committee Member

_____ Dr. Jill Nelson, Committee Member

_____ Dr. Duminda Wijesekera, Committee Member

_____ Dr. Andre Manitius, Department Chair

_____ Dr. Kenneth S. Ball, Dean, The Volgenau School of Engineering

Date: _____ Spring Semester 2013
George Mason University
Fairfax, VA

Multipath Routing and Load Sharing Using Game Theory

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Kunpeng Liu
Bachelor of Engineering
Tsinghua University, Beijing, China, 2005

Director: Dr. Bijan Jabbari, Professor
Department of Department of Electrical and Computer Engineering

Spring Semester 2013
George Mason University
Fairfax, VA

Dedication

This is dedicated to my loving wife Huijin, my two beautiful daughters Xiran and Xiyue.

Acknowledgments

I would like to thank the many friends, relatives, and supporters who have made this happen. My advisor, Dr. Bijan Jabbari, mentored me through the whole study. My friend, Dr. Stefano Secci, provided valuable comments on my research. My loving wife, Huijin, supported me in my study. My daughters, Xiran and Xiyue, taught me the meaning of life and joyfulness. My other committee members, Dr. Shih-Chun Chang, Dr. Jill Nelson and Dr. Duminda Wijesekera, were of invaluable help. My labmates, Dr. Akram Baharlouei and others, created a friendly and cooperative atmosphere in the lab.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AS.....Autonomous System
BGP.....Border Gateway Protocol
BGP-LS.....BGP Link State
DNS.....Domain Name System
ECMP.....Equal Cost Multipath
EL.....Entropy Label
ELI.....Entropy Label Indicator
ETR.....Egress Tunnel Router
IGP.....Interior Gateway Protocol
IS-IS.....Intermediate System to Intermediate System
ITR.....Ingress Tunnel Router
LAG.....Link Aggregation Group
LER.....Label Edge Router
LISP.....Locator/Identifier Separation Protocol
LSP.....Label Switched Path
LSR.....Label Switch Router
MP-BGP.....Multi-Protocol BGP
MPLS.....Multi-Protocol Label Switching
MPTCP.....Multipath Transmission Control Protocol
NE.....Nash Equilibrium
OSPF.....Open Shortest Path First
PCE.....Path Computation Element
TE.....Traffic Engineering
TL.....Tunnel Label
VPN.....Virtual Private Network

Abstract

MULTIPATH ROUTING AND LOAD SHARING USING GAME THEORY

Kunpeng Liu, PhD

George Mason University, 2013

Dissertation Director: Dr. Bijan Jabbari

The problem of multipath routing has received a considerable amount of attention due to its broad application in mitigating routing issues, such as addressing the problem of traffic distribution across a set of resources for which individual paths may not have the capacity to carry the load, achieving a higher resiliency through active backup paths, facilitating reliable network operation by routing and resource management, and meeting performance measures. Such problems can be formulated as a bimatrix game using game theory with different objective functions.

This dissertation studies how to solve a general multipath routing problem using game theory. Through studying and categorizing the costs for source and destination nodes, the interaction between the two distant independent nodes can be modeled as a non-cooperative game. When only considering single metric, the game can be further expressed as a cardinal potential game. Not only does this characteristic simplify the process of finding the Nash Equilibria (NEs) but also brings a multipath load sharing framework by using the potential value to evaluate the routing strategies.

In addition, a novel vectorized routing cost model, based on vector space and game theory, is defined to overcome the limitation of the previous model. The vectorized model provides the ability of considering multiple metrics simultaneously. To solve the vectorized routing model, a set of universal refinement tools is proposed through analyzing the rationale behind the behaviors of nodes under the context of game theory. In particular, one of refinement tools is proved as the extensive form of the potential value method. Through the refinement tools, a generalized multipath load sharing framework is achieved, which is applicable to more general settings since it does not depend on specific characteristics of the game.

Multi-criteria simulations on real instances show that significantly higher routing resiliency can be achieved through the generalized multipath load sharing framework.

Chapter 1: Introduction

The Internet has been evolving from an academic network managed and operated by researchers, to a worldwide and ubiquitous network interconnecting thousands of independent networks and potentially billions of users. Following its expending, the Internet plays a more and more important role in peoples' daily lives as well as works, thus its functioning statuses, e.g., its bandwidth, resiliency, security, etc., are vital to internet service providers.

In this chapter, an abstract model of general multipath routing is proposed, and an understanding to the model through *game theory* aspect is presented. With the tools, brought by game theory, the multipath routing problem can be solved mathematically.

1.1 Background

The Internet can be defined as a physical interconnection of Autonomous System (AS) networks, and its service consists of allowing end-to-end data transmissions between hosts. Internet hosts are identified by IP addresses and exchange packets of information that are then routed asynchronously across the networks of different carriers and routers of different vendors, where packets are independently switched from an ingress link to an egress one.

At the host-end side, packets are sent in the context of TCP or UDP session flows. Both TCP and UDP identify the related applications via connection ports, and TCP additionally offers congestion control mechanisms to allow bit-rate shaping in case of packet loss or detected congestion alarms. The congestion prevention and reaction behaviors of TCP are indirectly influenced by router's input and output queues that shape the traffic rate.

The IP address assigned to a host does not only uniquely identify the host across the

Internet, but also localizes the network the host belongs to. To instantiate an Internet connection, often the two end hosts do not know the respective IP addresses, which are in fact resolved by querying (with the other's host name) the Domain Name System (DNS) protocol architecture.

IP packets contain source and destination IP addresses and ports, and their Internet network path depends on two protocols, the Border Gateway Protocol (BGP) and an Interior Gateway Protocol (IGP). The first one selects the list of AS networks that the packet will cross on the way to its destination, the latter one determines the router-level path inside each AS network, from the ingress point toward the egress one or the destination. One single Inter-AS protocol (BGP) is adopted in the whole Internet, while many IGPs (e.g., OSPF, IS-IS) exist.

The Inter-AS route is selected by the BGP decision process [1]. When multiple AS paths to a destination network prefix are available, a cascade of criteria is employed to compare them. The first one is the "local preference", which is local policies, mainly guided by economic issues, e.g., a peering link (i.e., free transit) is preferred to a transit link (transit fees). Marking routes with local preferences, an AS can thus implement peering and transit settlements. The subsequent BGP criteria purely incorporate operational network issues: smaller AS hop count, smaller MED (Multi-Exit Discriminator, a metric used by neighbor ASes only to discriminate among upstream links), closer egress point (also called "hot-potato", external BGP speakers, or EBGP, are preferred to internal ones, or IBGP, and shortest internal routes are preferred to longer ones), and possibly other vendor specific rules. If not enough, the AS path learned by the router with the smaller IP is selected (rule also called "tie-breaking"). Considering these criteria, BGP selects the best AS path which is then advertised to the neighbors (if not filtered by local policies).

BGP relies on a flat routing mode using path vectors for each IP network prefix, announced independently in an uncoordinated fashion. Meanwhile, IP prefix de-aggregation is very popular among most ASes [2] due to their incentives to control inbound traffic and also to

avoid prefix hijacking(security countermeasure against unauthorized prefix announcements). Therefore, the flat routing model of BGP routing faces a big challenge in scalablility for such a large number of networks.

A new protocols, called Locator/Identifier Separation Protocol (LISP) [3], is proposed to tackle the Internet routing inscalability and other issues. These objectives are pursued by a new addressing context for the Internet architecture in which the two functions currently absolved by the IP address, that is, the network location and the host identification, are separated. The idea is to assign to every Internet host an identifier which will not be used to route packets, but simply to identify the host. IP routing locators will indicate the way from the source network to the destination network through the Internet.

There are basically two ways to separate locators from identifiers, one way is host-based and the other one is network-based. In host-based solutions, the hosts manage the mapping between locators and identifiers, while in network-based solutions this operation is done by the network at some gateways between source-destination pairs. Host-based solution, on the contrary of network-based, do not reduce the dimensions of the core routing tables.

It is easy to understand when we compare network-based LISP with the normal post-mail geolocalization: instead of using the destination person's name to route letters, in the core post routing centers only the destination's state or metropolitan area is used, and the final address and the person's name are only used in the last miles. Similarly, in a LISP-based Internet, routing locators are used to route the packets across the core of the Internet, and the packet identifier is only used in the edge networks to route the packet to the destination host.

Since locators have great potential for address aggregation in network-based LISP, the dimensions of the core routing tables can shrink dramatically if network-based LISP can be applied. Then, the scalability issue would be solved fundamentally.[1]

[1]The details are discussed in chapter 6.

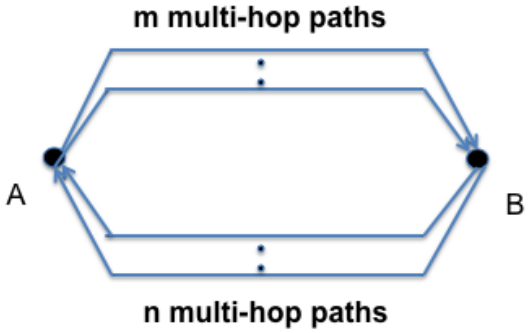


Figure 1.1: A general representation of multipath routing

1.2 Statement of the Problem

Given the infrastructure of the Internet, there are a number of reasons to use multipath routing and load sharing in Internet. These include addressing the problem of traffic distribution across a set of resources for which individual path may not have the capacity to carry the load, achieving a higher resiliency through active backup paths, facilitating reliable network operation by routing and resource management, and meeting performance measures.

Consider two nodes A and B, as depicted in Fig. 1.1, are connected by m and n paths in the directions A to B and B to A, respectively. Each path is assumed traversing across a multiplicity of hops. For each bi-directional path between nodes A and B (A to B and B to A paths), we consider the associated metric for each node to be a function of costs of the two paths in each direction. The costs can represent the attributes of the path such as traffic load, delay, etc. The objective functions for nodes are to minimize the total costs associated with the bi-direction traffic transmission. Based on the objective functions, we study how to find the set of rational solutions involving the percentage usage of each path for each node A and B.

1.3 Game Theory Approach

Since the associated metric for each node is a function of costs of both directions, the cost of one node is not only decided by its own strategy, but also affected by the strategy of the other node. In this sense, the behavior of node A and B can be modeled as a bimatrix game using *Game Theory* with certain objective functions.

Game theory is a powerful tool that has been applied in several engineering applications, notably in network routing, which is usually referred to as network routing game. The idea behind the prevalence of game theory in the area of network routing seems that the players in network routing, e.g., telecom firms, ISPs, or even Internet users, seek to benefit themselves first when they are making strategies, meanwhile their benefits are also affected by others' strategies. Under such a scenario, each player needs to consider other players' influence while optimizing his own criterion, and game theory provides a set of mathematical tool to analyze such complex interactions among rational players.

Depending on the objective functions, the Nash equilibrium (NE) may be the solution for certain cases. In general, however, the existence of a pure NE for the game can not be guaranteed, and, therefore, it is necessary to extend the concept of NE to also include mixed NE in order to analyze for solutions. A mixed strategy is a probability distribution vector over all possible strategies, and it represents the probabilities that a player chooses his available pure strategies. A mixed NE is a point that no player can improve his payoff by unilaterally changing his probability vector over all his possible strategies. In the extended concept, the existence of a NE is guaranteed for finite noncooperative games by Nash's Theorem [4].

1.4 Outline of the Dissertation

The dissertation is organized as follows. A relatively concrete multipath routing model is raised in chapter 2, followed by an overview of the related works, which includes selected

existing and emerging techniques that may be applied to solve the multipath problem. A preliminary model is presented in chapter 3, starting with a simple game, then arising the game properties and proposing the entire model. A novel vectorized routing cost model as well as a generalized multipath is proposed in chapter 4 to overcome the limitation of the previous model. Certain applications are discussed in chapter 5. The Internet hierarchical interconnection is analyzed in chapter 6. The dissertation is concluded in chapter 7.

Chapter 2: Application of Model to Multipath Traffic Engineering and Routing

Routing represents the process of selecting network paths under certain criteria for the Internet traffic to pass through. Traffic engineering is a series of techniques that deal with the issue of performance evaluation and performance optimization of operational networks, and it encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic [5].

Nowadays, certain TE techniques have been applied in utilizing multipath routing and load sharing, e.g., Link Aggregation Group (LAG) and Equal Cost Multipath (ECMP). LAG refers to various methods of aggregating multiple network links in parallel to increase the overall throughput, and to provide redundancy in case one of the links fails. ECMP allows load sharing over several shortest paths between two nodes separated by one or more hops in the network in case that multiple shortest paths could be achieved (in the cost metric selected for the routing). ECMP load sharing can be applied per packet, however may result in Jitter or delay and even Out-of-Order packets to ultimate destination. Current ECMP load sharing is flow specific, especially for Label Switched Paths (LSPs) within Multiprotocol Label Switching (MPLS) [6] enabled networks. As an advanced forwarding scheme, MPLS extends routing in respect of packets forwarding and path controlling with labels. Usually, MPLS labels are pushed by the ingress Label Edge Router (LER), referred and/or swapped by Label Switching Routers (LSRs), and popped by the egress LER. Sometimes, LSPs can also push and pop labels to assist the packets forwarding. The forwarding path, which is controlled by labels with starting from the ingress LER and ending at egress LER, is called LSP. The proposed approach in this dissertation has the potential to become a flow specific TE mechanism, which can be based on MPLS TE.

To solve the multipath routing problem, it is necessary to have a detailed model to work on instead of the abstract one. In this sense, a relatively concrete multipath routing model is raised in chapter, followed by an overview of the related works, which includes selected existing and emerging techniques that may be applied to solve the multipath problem.

2.1 A Multipath Routing Model

Assume that A and B are *edge nodes* (can be interpreted as routers in most cases) that directly provide Internet service to the customers in a network with a large topology, and A and B need to exchange a relevant amount of traffic in a stable manner. Within such a large topology, it is not practical to assume that every node is aware of all the details about the network. The challenge is how should A and B achieve the set of rational solutions involving the percentage usage of each path for each node A and B.

Some additional assumptions are made to further analyze the problem:

1. certain metric can be utilized to evaluate the resiliency of a link or path, and the metric value can be treated as transit cost, which bears the meaning that the lower the value of the link or path is, the higher the resiliency of the link or path is;

2. every node has full knowledge of its neighbor nodes which help it to communication with other nodes;

3. the large topology of the network is due to the fact that the network is providing Internet service to a tremendous large number of costumers that geographically locate in a lot of areas;

4. edge nodes take a very large portion of the total number of the routers, and the other part of the nodes, called *transit nodes*, constitute the Infrastructure of the network, which means that transit nodes are capable to know the details about the Infrastructure of the Internet.

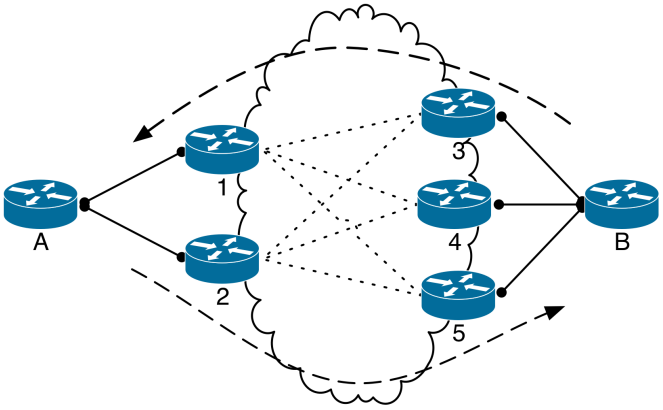


Figure 2.1: Edge-to-edge routing interaction example.

In fact, the transit nodes directly connecting with edge nodes have two functions: gateways for the source nodes and locators for the destination nodes. The two roles are going to be distinguished in the analysis to avoid possible mislead.

In addition, each edge node has its own preferences towards its gateways and locators, and such behavior can be observed from BGP local preference and AS path prepending. Suppose the preferences is measured by preference cost, with the interpretation that the lower the cost is, the higher the preference is.

An illustration is shown in Fig. 2.1. In Fig. 2.1, node A and B are edge nodes that directly provide Internet service to the customers, and nodes I, $I \in \{1, 2, 3, 4, 5\}$ are transit nodes. A knows node 1 and 2 can be used to communicate with B as well as the corresponding costs, and vice versa. Besides, nodes I, $I \in \{1, 2, 3, 4, 5\}$ have the full acknowledgement about the transit nodes connecting situation.

It is worth stressing here that, as a matter of fact, the additional assumptions made above are quite practical. Take the whole Internet as an example, even though the Internet consists of a very large number of AS networks, about 97% of them directly connect to the customers all over the world, and only 3% AS networks constitute the Infrastructure of the Internet [7]. The details are analyzed in chapter 6. In this sense, it is totally practical that the nodes constituting the Infrastructure of the Internet are aware of the connecting

status.

Suppose A and B are aware of each other's transit nodes as well as the corresponding costs through other communication channels or some mapping methods, they hope to achieve a relative low cost routing strategy for themselves using multipath routing and load sharing.

This dissertation presents a generalized multipath routing framework to improve the routing and load sharing resiliency. Based on the getaways and the locators for the source and destination nodes as well as the corresponding preferences, the routing problem can be modeled with *Game Theory*. The resolution of the game can provide a traffic distribution strategy for the two nodes.

2.2 Related Work

The resiliency of the Internet has attracted numerous research efforts, and valuable and diverse directions have been followed to improve or even re-design the Internet. This section will give an overview of related work from three aspects, including the work on TE, multipath routing as well as applying Game Theory on network routing, which is usually called *network routing game*.

2.2.1 Traffic Engineering (TE)

At present, IP packet routing relies on two types of protocols, the Border Gateway Protocol (BGP) and Interior Gateway Protocols (IGPs). The first one selects the list of AS networks that the packet will cross on the way to its destination, while the latter determine the router-level path inside each AS network, from the ingress point towards the egress point or the destination. One single Inter-AS protocol (BGP) is adopted in the whole Internet, while many IGPs (e.g., OSPF, IS-IS, RIP) exist.

On inter-AS routing level, the current BGP protocol offers local preference and AS path prepending to control the direction and the load of outbound and inbound traffic toward and from upstream AS networks. The local preference can be assigned to rank upstream AS networks in order to control the outbound traffic, while with AS path prepending one AS network can artificially increase the AS path length by repeatedly announcing its own AS number to distract its inbound traffic toward certain path [8] [9]. Nevertheless, AS path prepending works on a try-and-hope fashion, as other AS networks can easily bypass this effort by filtering out the AS path prepending information.

On intra-AS routing level, IGPs regularly apply IGP link weights optimization to minimize the maximum link load in case of traffic matrix variations, link and node failures [10] [11]. In addition, thanks to the adoption of the Multi-Protocol Label Switching (MPLS) [6], which simplifies forwarding by means of pre-binded labels, TE extensions can be adopted to perform constrained shortest path routing. For instance, by allowing more than one link metric in link state protocols, it is possible to account for additional performance factors, e.g., propagation delay, link capacity and failure probability [12].

Following the research on Internet routing, many valuable and diverse novel TE mechanism and techniques are emerging recently, most of which seem have the potentials to prompt the capacities of TE to a certain extent.

Flow-based switching is an interesting direction recently attracting much attention [13]. Performing data switching using transport-level information, let it be TCP or UDP flows, rather than only network-level (IP routing) information (as currently done), the routers will have the intelligence to theoretically grant improvements on the perceived QoS, not only for streaming audio or video applications, but also for elastic transfer of digital content (downloads, webpages, email, etc). These improvements can be reached by allowing connection admission controls (CACs) for (transport-level) application flows – hence not for source-destination router/network pair aggregate (network-level) flows. The flow shall also be classified with respect to its characteristics (streaming or elastic, video or voice, bulk

transfer or web browsing, etc) so as to assign it an appropriate output queue. In case of congestion, current IP switching will drop packets irrespective to which flows they belong to, which will cause retransmissions driven by TCP congestion control. With flow switching CAC, once flows are accepted, resources are booked for them; then, if saturation is reached, only packets from new flows are dropped, which does not decrease the performance for already admitted flows.

Inter-AS MPLS is another technique that is under discussion nowadays. At the IETF, standardization efforts have been aimed, in the last five years, to the extension of the MPLS protocol beyond the AS boundaries, i.e., at the Internet scope. These efforts brought the definition of the inter-AS MPLS extension. The inter-AS MPLS protocol architecture also include the so-called Path Computation Element (PCE) architecture (see [14]) to allow cross-AS path computation with topology abstraction and routing confidentiality (a requirement of paramount importance for inter-AS routing). Nevertheless, the protocol is being implemented mostly for MPLS-VPN services, that do not need path computation but only inter-AS label binding between two neighbor ASes, possibly in cascade. Indeed, Inter-AS MPLS is not intended as a Internet-wide protocol solution; just the requirement of having an MPLS network practically excludes most of the stub ASes (those at the border of the Internet, about 85% of all the ASes). The PCE architecture being implemented in MPLS networks is mainly for intra-AS multi-domain or multi-area path computation, despite numerous testbeds at the multi-AS scope are being deployed successfully (see, e.g., [15]). Real implementations of inter-AS MPLS in operational networks may be thus limited to a few AS networks sharing common service provisioning and traffic engineering interests, i.e., to some Internet core carriers and operators.

Meanwhile, the routing working group of the IETF has been conducting standardization activities to add TE functionalities to inter-AS MPLS, which is called inter-AS MPLS-TE [16]. These extensions also encompass new versions of the underlying routing protocols

defined as OSPF-TE, IS-IS-TE and inter-AS RSVP-TE. The purpose of inter-AS MPLS-TE is to allow for explicit constrained inter-AS routing, and there have also been successful testbeds about this type of extensions coupled with the PCE architecture [17]. These works about inter-AS MPLS-TE open the way toward a connection-oriented paradigm for the Internet architecture, which at some extent is opposed to the first Internet ideas of a control-plane free and state-less Internet routing.

2.2.2 Multipath Routing

To run multiple instances of link state routing protocol is an approach to enforce multi-path routing[18]. In this circumstance, each link has a vector of weights, which can be decided independently with each other to customize the different shortest path for different applications, e.g., one weight is for delay and one weight for throughput. Therefore, multi-path routing can be naturally introduced for the same pair of source and destination node according to the weights vector for each link.

Multipath Internet routing could also be applied with BGP at the inter-AS level, and major vendors' routers have already implement the multipath mode [19] [20]. However, recent analysis shows that BGP multipath is practically unused at present [21]. One reason behind its low adoption resides in the fact that it would add uncertainty to interconnection agreements, which are based on aggregate volumes estimates. Another important reason is that indiscriminate BGP multipath routing can have counter-effects on routing stability and convergence if all available paths are indiscriminately chosen. Instead, the choice on the multipath routing solution should be more selective as, e.g., suggested in [22] that the multiple paths are chosen on the basis of routing equilibria.

In addition, when multiple ASes are cooperative with each other, multipath routing can be implemented by deflection[23]. The source node can explicitly enforce multipath routing by encapsulating packets and sending to a deflection node, which lies on an alternative path.

The concept of multipath routing can also be implemented on TCP layer. Research efforts in some European projects and at the IETF are pushing toward the standardization of a multipath mode for TCP, called Multipath TCP (MPTCP) [24]. The basic idea is to inform the host about the multi-homing configuration of the AS network it belongs to, i.e., the different available gateways and therefore the different loose paths toward the destinations. Managing one state per available path, when loss is detected on one path MPTCP dynamically balances the load towards the other(s). With MPTCP, a host would have higher degree of freedom to control its own traffic, and control-plane functionalities would be moved from the network to the host.

2.2.3 Network Routing Game

Game theory is a powerful tool that has been applied in several engineering applications, notably in network routing, which is usually referred to as network routing game. The idea behind the prevalence of game theory in the area of network routing seems that the players in network routing, e.g., telecom firms, ISPs, or even Internet users, seek to benefit themselves first when they are making strategies, meanwhile their benefits are also affected by others' strategies. Under such a scenario, each player needs to consider other players' influence while optimizing his own criterion, and game theory provides a set of mathematical tool to analyze such complex interactions among rational players.

The models in game theory can be roughly divided into two categories, non-cooperative games and cooperative games. Non-cooperative games model the interactions among competing players, where each player chooses its own strategy independently for improving the utility or performance for itself. Several concepts exist for solving non-cooperative games, e.g. *Nash equilibrium* (NE). While non-cooperative games refer to the competitive scenario, cooperative games analyze the behavior of rational players when they cooperate with each other. Players in cooperative games will choose to be cooperative in order to improve the coalition's performance if they can also improve their own performance at the same

time. Since the applications of cooperative games in network routing games are sparse, the following review will only focus on non-cooperative routing games.

The simplest model is the parallel paths routing game, where each player has a given amount of flow to ship and has several parallel paths that he can split the flow into. This model was studied in detail in [25] and [26]. It was shown that the NE of such systems is usually unique under reasonable convexity conditions. Moreover, the monotonicity properties of the NE for flows were also characterized in an intuitively appealing way. The conclusions, however, cannot be guaranteed in a more general network topology without additional assumptions. A discrete model with a general network topology was presented in [27], with the assumptions that each player has exactly one unit to ship and the path cost is additive with respect to every link cost, the existence of NE can be proved. Nevertheless, if a player has more than one unit to transit, the equilibrium may not exist.

It is worth noting that additional restrictions can also be applied in this type of games, such as the QoS, including the delay as well as other criteria [28]. The work is improved in [29], where the authors consider the case of multiple flows with QoS constraints over a multipath network. In this QoS game, the players are the flows, each player's strategy space is the percentage of utilization of the high quality path, and the utility function for each player is the quality of the throughput of each flow. They analyze the existence and uniqueness of a NE, and demonstrate that the NE solutions are optimal in maximizing the successful transmission of individual streams with numerical experiments.

In [30], the authors considered how to minimize the aggregated latency when assigning traffic to the links of the network, given the lack of coordination among the nodes of the network. In particular, they assumed that that every node was selfish and selected the link with the minimum latency from its own perspective. They quantified the influence of unregulated traffic to network performance, and proved that the total latency of the routes chosen by selfish network users is at most 4/3 times the minimum possible total latency when the latency of each edge is a linear function of its congestion. Further work in [31]

showed that the price of anarchy does not depend on the network topology and the worst possible ratio may occur even in very simple networks.

Some non-cooperative games are designed to model flows for *elastic demand* whose total amount of flow to route on the network is set optimally, simultaneously with the routes, with reference to the network characteristics as well as the demand function.

In [32], the authors studied how to share available bandwidth within a large-scale broadband network between competing streams of elastic traffic. Given the additive increase/multiplicative decrease schemes, they proposed an optimization framework, which decomposed the overall system problem into separated problems in a stable manner.

In [33] and [34], it is assumed that the route costs are the sum of the cost of the constituent links. Then the resolution for the routing and flow control models will be considerable simplified; meanwhile it is possible to use shortest-path algorithms to solve the sub-problems. As a matter of fact, when the interactions among users are fixed, the routing and flow control problems can be purely expressed as shortest path problems.

In order to maximize the average network throughput under the constrain of the average network delay, a Markov queuing model is presented in [35] to find the optimal decentralized flow control mechanisms under the network optimization criterion which can also be understand as a non-cooperative game. The existence of the equilibrium for the game is analyzed in [36].

The peering interconnections between different carriers can also be modeled with non-cooperative games. In [37], a non-cooperative coordinate framework is presented to mitigate the lack of coordination between providers. With considering both routing cost and congestion cost, they provide peering equilibrium multipath coordination polices, which can decrease the routing cost around 10%. The previous model is improved in [38] to also take into account the occurrence of potential impairments, e.g., traffic matrix variations, intra-AS and peering link failures. Simulations show that the resilient peering equilibrium

multipath polices can adaptively prevent from peering link congestions and excessive route deviations after failures.

Non-cooperative repeated games are another type of games to capture the impact of a player's current action on the future actions of other players. In [39], communication networks shared by selfish players are modeled as non-cooperative repeated games to investigate the existence of a NE that achieves the system-wide optimum cost. It is proved that the NE, which achieves not only the system-wide optimum cost but also a cost for each player no greater than its stage game NE cost, always exists for two players multiple paths networks. The conclusion can be extent to n players as long as the players share the same source and destination nodes. Nevertheless, for more general cases that multiple players have multiple pairs of source and destination nodes, such a NE may not exist.

Chapter 3: Preliminary Solution to Multipath Routing Using Game Theory

Suppose that two edge nodes exchange a relevant amount of traffic in a stable manner in a large topology network. Due to the large topology, they are not aware of every detail of the network. If we assume that the transit cost and the preference cost are in the same unit or can be translated into the same unit, which is termed as routing cost, then the interaction of the two nodes can be modeled with a potential game, which will bring us a multipath routing framework to improve the Internet routing resiliency. We start with a simple game, then arise the game properties and present the completed model. To illustrate the model clearly, the following explanations are based on Fig. 3.1, but the model can be applied to any topology network as long as the network follows the assumptions mentioned previously.

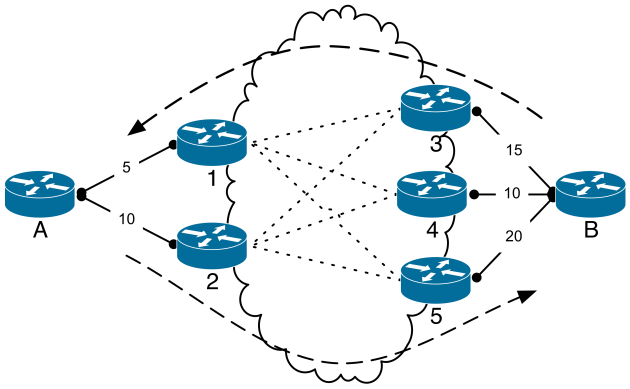


Figure 3.1: Edge-to-edge routing interaction example

Table 3.1: A locator routing game.

A \ B	L_1	L_2
L_3	**5,15**	**10,15**
L_4	**5,10**	**10,10**
L_5	**5,20**	**10,20**

3.1 An Introductory Scenario

Suppose A and B have acknowledged each other's transit nodes as well as the corresponding preference through some mapping methods or other communication channels, a straightforward but naive way is to follow the other party's interest, such as the source chooses the locator following the announced destination's preferences (e.g., minimizing its routing cost); this would be strategically acceptable in the cases that the two edge nodes belong to the same AS authority, or to two strategically dependent ASes (belong to the same company or dependent companies).

This dissertation focuses, instead, on a non-naive context in which the two edge nodes are independent and normally act following their own preferences first. In such a scenario, their strategic routing interaction can be modeled with non-cooperative game theory. Table 3.1 shows the locator routing game setting in strategic form corresponding to the scenario in Fig. 3.1, where the list of strategies available to A corresponds to the three locator-getaways of B (and conversely). In Table 3.1, the strategies are noted as L_i, where i index the locator nodes. Each possible strategy profile indicates the cost for A on the left and that for B on the right, accounting for the cost that each player's decision impacts on the other player, i.e., the locator cost. The profile (L_4, L_1), e.g., corresponds to the routing solution traced in Fig. 3.1.

All the profiles in Table 3.1 are (pure-strategy) NEs, i.e., for each player there is no preference over the available strategies. Indeed, the game is a dummy game, which highlights that using the destination's locator preferences without a traffic engineering purpose would

Table 3.2: Joint routing game

A \ B	G_3L_1	G_3L_2	G_4L_1	G_4L_2	G_5L_1	G_5L_2
G_1L_3	10,30	15,30	**10,25**	**15,25**	10,35	15,35
G_1L_4	10,25	15,25	***10,20***	**15,20**	10,30	15,30
G_1L_5	10,35	15,35	**10,30**	**15,30**	10,40	15,40
G_2L_3	15,30	20,30	15,25	20,25	15,35	20,35
G_2L_4	15,25	20,25	15,20	20,20	15,30	20,30
G_2L_5	15,35	20,35	15,30	20,30	15,40	20,40

be a routing practice rationally lack of motivation. Therefore, it is of importance to define coordination mechanisms to take benefit from the novel traffic engineering capabilities coming from the getaways and locators. In fact, the introduction of locators for destination nodes brings a larger path diversity into Internet routing, which can undoubtedly increase the overall resiliency.

3.2 Coordinated Joint Routing

It is possible for the two nodes agree in jointly routing their traffic under certain coordination equilibria if a proper game model can be proposed. This requires that the routing strategy of one node will not only affect the routing cost of the other node as shown in Table 3.1, but should also affect the routing cost of itself. Therefore, the strategy of one network is extended as the combination of its egress gateways as well as the locators of the destination. It is assumed (for the moment) that the locator preference applies also as a *gateway preference* for the egress direction, and the routing cost for one network is the sum of its gateway cost and its locator cost, as shown in Table 3.2 for scenario of Fig. 3.1.

In Table 3.2, the strategies are noted as G_iL_j, where i and j index the gateway nodes and the locator nodes. Now the decision is not only about the locators of the destination, but also about its egress gateways; e.g., G_1L_4 is a strategy for A that suggests to route the flow across getaway 1 toward locator 4 on the way for the destination. Table 3.2 indicates in

bold the six NEs of the corresponding routing game. For the sake of clarity, (G_1L_5, G_4L_2) is a NEs and the equal-cost (G_2L_3, G_3L_1) is not because for the first both the players have no incentive to change their strategies – for A, G_2L_x strategies have a cost of $15 > 10$ and the remaining strategies have an equal cost, for B G_3L_x and G_5L_x have a cost higher than 30, and equal to for the remaining strategies – while for the latter both have incentives to change to a strategy with a lower unilateral cost.

Among the six (pure-strategy) equilibria of Table 3.2, the one in italic (G_1L_4, G_4L_1) is the efficient one (more precisely, Pareto-superior to the others, see Appendix. C), i.e., one passes from any other equilibrium to it without increasing the cost for any player, but decreasing the cost for at least one.

It is worth noting that under the presented model, it is possible to achieve multiple equilibria even after applying the Pareto restriction. More precisely, this can happen in the case of equal gateway/locator preferences for all or a subset of the available gateway/locators at each side. The selection of multiple equilibria implies, in fact, a form of multipath routing with selective load sharing towards a subset of the available locators for the destination.

3.3 Setting with Forward Route Costs

An assumption taken so far is that the locator cost is equal to that of the gateway, i.e., the same routing cost is considered for both the upstream and the downstream flows. A more realistic assumption is that these two costs are different to each other, since the Internet routing is usually asymmetric due to routing policies. In addition to this, the cost of forwarding between transit nodes is also added into consideration. In this way, the game is slightly changed, with an ingress cost for the locator, and an egress cost for the forward route. The latter can be seen as sum of a gateway cost, generally different from the locator cost, and a transit path performance-evaluation cost. Therefore each edge node needs to

consider both the complete forward route costs and the ingress costs from locators for the backward flows (whose route is unknown to them).

Different methods can be conceived to estimate the transit path performance-evaluation cost. One can use rude yet efficient methods such as the node hop count, or one can map in the cost monitored performance along a route to assess its resiliency. Moreover, this may be done locally in the node or externally in a ranking middle box server (made available also by other entities than the providers) as discussed in [40], where a review of possible path ranking techniques is also given. In this dissertation, a novel metric is defined to take into account both the node hop count and the paths diversity with the idea that the more paths are available, the more resilient the transit route is; in case of failure along one path, alternative paths shall be available to the gateway nodes.

Let $\Omega_{i,j}$ be the set of available paths between a gateway i and a locator j, and let $L(\omega)$ be the node hop count of the path $\omega \in \Omega_{i,j}$. We believe it is appropriate to model the set of paths along a node as a system of resistors in parallel, where a resistance corresponds to a path length, and and the equivalent resistance (L_{eq}) can be computed. Lengthy paths bring more negligible contributions, and the more available paths the lower route cost. In the following, $c_{i,j}$ represents the transit path costs from the source toward the destination passing by the source's gateway i and destination's locator j, then:

$$c_{i,j} = \lceil A \cdot L_{eq} \rceil = \left\lceil A\Big(\sum_{\omega \in \Omega_{i,j}} \frac{1}{L(\omega)}\Big)^{-1} \right\rceil \tag{3.1}$$

where A is an arbitrary constant.

Fig. 3.2 shows an example for (3.1) with fives paths available where the lengths for the first four are 2, 3, 4 and 5 and the X-coordinate represents the length of the fifth one. Certainly, other cost functions can be conceived, and functions for different players can be different to each other; it is worth noting, however, that in order to maintain the good game properties

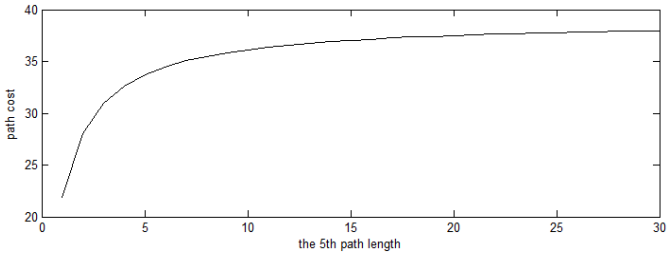


Figure 3.2: An example of the path cost function for $A = 50$

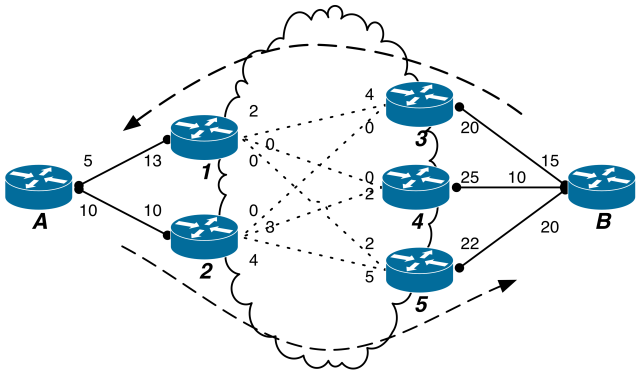


Figure 3.3: Edge-to-edge routing interaction example with forward path costs

explained hereafter, the different player functions have to be independent of each other.

In Fig. 3.1, the following settings are considered:

$c_{1,3} = 17, c_{1,4} = 13, c_{1,5} = 15, c_{2,3} = 10, c_{2,4} = 12, c_{2,5} = 15$

$c_{3,1} = 22, c_{3,2} = 20, c_{4,1} = 25, c_{4,2} = 28, c_{5,1} = 22, c_{5,2} = 26$

Hence, Fig. 3.3 as well as Table 3.3 (the exponent meaning is explained in Sect. 3.5) can be obtained with a single NE.

3.4 Mathematical Notations

The routing game can be described as $G = (X, Y; f, g) = G_s + G_d$, sum of a selfish game and a dummy game, respectively; let f and g be the cost functions, and X and Y the strategy

Table 3.3: Bidirectional routing game with forward path costs

A \ B	G_3L_1	G_3L_2	G_4L_1	G_4L_2	G_5L_1	G_5L_2
G_1L_3	22,37 $^{(5)}$	27,35 $^{(3)}$	22,40 $^{(8)}$	27,43$^{(11)}$	22,37 $^{(5)}$	27,41$^{(9)}$
G_1L_4	*18,32* $^{(1)}$	23,30 $^{(-1)}$	18,35 $^{(4)}$	23,38 $^{(7)}$	*18,32* $^{(1)}$	23,36$^{(5)}$
G_1L_5	20,42 $^{(3)}$	25,40 $^{(1)}$	20,45 $^{(6)}$	25,48 $^{(9)}$	20,42 $^{(3)}$	25,46$^{(7)}$
G_2L_3	15,37$^{(-2)}$	**20,35**$^{(-4)}$	15,40 $^{(1)}$	20,43 $^{(4)}$	15,37$^{(-2)}$	20,41$^{(2)}$
G_2L_4	*17,32* $^{(0)}$	22,30 $^{(-2)}$	*17,35*$^{(3)}$	22,38 $^{(6)}$	*17,32* $^{(0)}$	22,36$^{(6)}$
G_2L_5	20,42 $^{(3)}$	25,40 $^{(1)}$	20,45 $^{(6)}$	25,48 $^{(9)}$	20,42 $^{(3)}$	25,46$^{(7)}$

sets, of edge node A and B, respectively. Each strategy $x \in X$ or $y \in Y$ indicates the source gateway and the destination locator. The strategy set cardinality is equal to the number of source gateways $\times$ the number of destination locators. G_s considers the forward path cost only, while G_d considers backward locator cost only, impacted by the other network's routing decision – we already discussed an example of dummy game in Table 3.1.

$G_s = (X, Y; f_s, g_s)$, is a purely endogenous game, where $f_s,\ g_s : X \times Y \to \mathbf{N}$ are the cost functions for A and B, respectively. In particular, $f_s(x, y) = \phi_s(x)$, where $\phi_s : X \to \mathbf{N}$, and $g_s(x, y) = \psi_s(y)$, where $\psi_s : Y \to \mathbf{N}$. For the game in Table 3.3, e.g., consider the profile $(\widetilde{x}, \widetilde{y})$ with $\widetilde{x} = G_2L_3$ and $\widetilde{y} = G_4L_1$; we have:

$f_s(\widetilde{x}, \widetilde{y}) = \phi_s(\widetilde{x}) = c_{2,3} = 10$

$g_s(\widetilde{x}, \widetilde{y}) = \psi_s(\widetilde{y}) = c_{4,1} = 25$

$G_d = (X, Y; f_d, g_d)$, is a game of pure externality, where $f_d,\ g_d : X \times Y \to \mathbf{N}$, $f_d(x, y) = \phi_d(y)$ and $\phi_d : Y \to \mathbf{N}$, $g_d(x, y) = \psi_d(x)$ and $\psi_d : X \to \mathbf{N}$. Let E be the edge link set, and let $c(l'_i)$ be the routing cost across the ingress link l'_i by provider/locator i, with $l_i, l'_i \in E$. For the above example:

$f_d(\widetilde{x}, \widetilde{y}) = \phi_d(\widetilde{y}) = c(l'_1) = 5$

$g_d(\widetilde{x}, \widetilde{y}) = \psi_d(\widetilde{x}) = c(l'_3) =15$

3.5 Pure-strategy Equilibrium Properties and Computation

$G_s + G_d$ is a cardinal potential game [41], i.e., the incentive to change players' strategy can be expressed with a single potential function (P) for all players, and the difference in individual costs by an individual strategy move has the same value as the potential difference. G_d can be seen as a potential game too, but with null potential. Hence, the potential $P : X \times Y \to \mathbf{N}$ depends on G_s only.

Generally, in non-cooperative games the NE existence is not guaranteed. As a property of potential games [41], the P minimum corresponds to a (pure-strategy) NE and always exists, however, the inverse is not necessarily true.

The exponents in the example of Table 3.3 indicate the potential values corresponding to the strategy profiles[1]. The NE is thus guided by G_s. The opportunity of using the minimization of the potential function to catch all the peering NEs represents a key advantage. It decreases the time complexity, which would have been very high for instances with many providers and locators. When there are multiple equilibria (possible with equal forward path and/or locator costs), G_d can help in selecting an efficient equilibrium in the Pareto-sense.

3.6 Enforcing Edge-to-Edge Load Sharing

In a getaway-locator routing framework, it is technically possible and desirable to implement *edge-to-edge load sharing* schemes. The presence of multiple locators for the same destination radically increases the Internet path diversity available to the source network. Moreover, with forward path ranking by the edges, load sharing is particularly desirable to avoid possible routing oscillations. In fact, if single path routing is used in the case that

[1] to explicate P in calculus an arbitrary starting potential has to be chosen; we set to 0 the potential of social welfare profiles, i.e., $P(x_0, y_0) = 0 \;\; \forall (x_0, y_0) \in X \times Y | f(x_0, y_0) + g(x_0, y_0) = \min\{f(x, y) + g(x, y)\}$.

multiple networks use the same path cost function and react synchronously to path performance degradation (assigning them higher costs), the single path is likely to suffer from performance loss in turn because of traffic overload, leading to possible persistent routing oscillations. A systematic yet fine-selected load sharing scheme can prevent these events affecting the Internet routing stability.

A generic way to implement load sharing is to assign a percentage weight to each route-strategy, indicating the distribution of egress traffic toward the destination along that route. Alternatively, a percentage weight can be assigned to the locators by the destination network as its desired distribution for the upstream network(s). Both ways are technically possible and somehow equivalent. We are thus interested in defining a method to set such traffic distribution weights that is strategically acceptable.

The selection of n multiple equilibria could result in an even load sharing distribution (at most $1/n$ load on each locator). Although acceptable, it is desirable to rank the equilibria following some rational criteria better considering the game dynamics so as to better meet routing stability requirements.

3.7 The Potential as an Equilibrium Refinement Tool

In our framework, the important question is: what is the strategically acceptable load sharing distribution technique for edge-to-edge flows? Theoretically, an answer to the question can be mixed strategy equilibria; however, for potential games they correspond to pure-strategy equilibria (see Appendix B).

In potential games, the potential value qualifies the profile sensibility and predict the behavior of the potential game [41]: the lower it is for a (equilibrium) strategy profile, the finer the profile is. However, the occurrence of multiple equilibria in G is not guaranteed - it happens only with equal egress and/or ingress costs[2] - and may be a rare event for small

[2]For the example of Table 3.3, multiple equilibria appear if $c_{1,4} = c_{2,3} = 10$, hence (G_1L_4, G_3L_2) as

instances; in these cases, load sharing could not be implementable.

Since load sharing is a key feature to improve Internet resiliency, it is desirable to increase the number of strategy profiles in the routing solution. The potential value can in fact help in extending the equilibrium set including also those profiles that are not pure-strategy equilibria, but that have good chances of becoming so in future settings. For example, in Table 3.3, the profiles having a potential equal to -2 have a good chance to become an equilibrium after slight changes of one or a few cost components; such profiles can be considered as better strategy profiles than other profiles with a higher potential.

With the aim to increase the path diversity of the routing solution, we can thus add those profiles that are not NEs but have very low potential values into the equilibria set and include them in the routing strategy. This corresponds to select the routing equilibrium within a profile set including all the strategy profiles with a potential value equal or below a pre-computed threshold (i.e., not only those with the minimum potential). Since the maximum and the minimum potential values change with the game configuration, the threshold can be set accounting for the statistical potential distribution. An acceptable threshold corresponds to the first quartile of the potential distribution. For example, in Table 3.3, the first quartile potential is equal to 1; therefore, the routing solution includes seven strategy profiles with a potential between -4 and 0. The threshold computation can, however, be adapted to the problem instances; for very large instances, more conservative threshold levels than the first quartile could be used.

A further implicit step that is rationally acceptable is to restrict the equilibrium set only to those that are not Pareto-inferior to any other selected equilibrium; in Table 3.3, this corresponds to discard (G_2L_3, G_3L_2) from the solution (even if it is the single pure-strategy equilibrium). Finally, we propose to use the potential values of the remaining equilibria to set the load sharing distribution, so that a lower potential value brings to a higher load ratio.

second equilibrium, or if $c_{5,1} = c_{3,2} = 20$, hence (G_1L_4, G_5L_1) as additional equilibrium; note that both the new equilibria are Pareto-superior to the incumbent one (G_2L_3, G_3L_2).

3.8 Load Sharing Distribution Computation

Let $\chi \in S_A \times S_B$ be the set of the profiles kept as solutions, τ be the threshold and $\mathbf{P}_{x,y}$ be the potential value of $(x, y) \in \chi$. Then,

$$\begin{aligned} \Lambda &= \sum_{(x,y)\in\chi} [1 + \tau - \mathbf{P}_{x,y}] \\ \Lambda_{\widetilde{x}} &= \sum_{(x,y)\in\chi | x=\widetilde{x}} [1 + \tau - \mathbf{P}_{x,y}] \\ \Lambda_{\widetilde{y}} &= \sum_{(x,y)\in\chi | y=\widetilde{y}} [1 + \tau - \mathbf{P}_{x,y}] \end{aligned} \tag{3.2}$$

where Λ represents the total weights, $\Lambda_{\widetilde{x}}$ (resp. $\Lambda_{\widetilde{y}}$) represents the weights for $s_{\widetilde{x}} \in S_A$ (resp. $s_{\widetilde{y}} \in S_Y$).

The traffic is assigned as:

$$\begin{aligned} p_{\widetilde{x}} &= \Lambda_{\widetilde{x}}/\Lambda, \quad \forall s_{\widetilde{x}} \in S_A \\ q_{\widetilde{y}} &= \Lambda_{\widetilde{y}}/\Lambda, \quad \forall s_{\widetilde{y}} \in S_B \end{aligned} \tag{3.3}$$

We can in this way fairly assign higher weights to those unilateral strategies that cover many solution equilibria. For example, in Table 3.3, we obtain the load sharing solution $b_{G_2L_3} = 8/16 = 50\%$ and $b_{G_2L_4} = 8/16 = 50\%$ for A, and $b_{G_3L_1} = 37.5\%$ and $b_{G_3L_2} = 25\%$ and $b_{G_5L_1} = 37.5\%$ for B. Note that without the Pareto restriction we would obtain, instead, $b_{G_1L_4} = 3/25 = 12\%$ and $b_{G_2L_3} = 15/25 = 60\%$ and $b_{G_2L_4} = 7/25 = 28\%$ for A, and $b_{G_3L_1} = 24\%$ and $b_{G_3L_2} = 52\%$ and $b_{G_5L_1} = 24\%$ for B, hence a more fragmented distribution.

To summarize, the routing solution is computed with the following steps:

1: build the game from the cost components;
2: compute the potential value vector of the game, its minimum and its potential threshold;
3: select all the profiles with a potential equal to or minor than the threshold;
4: apply the Pareto-restriction of the profile set; if empty, keep all the profiles;
5: compute the corresponding load sharing distribution for the remaining profiles.

Algorithm 1: Load sharing distribution computing steps

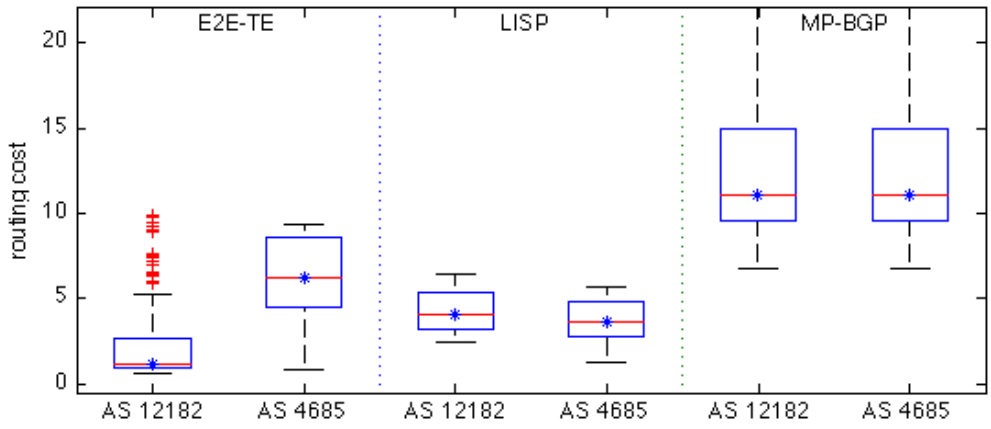


Figure 3.4: Boxplot statistics of routing cost for the solutions

3.9 Performance Evaluation

We simulated the edge-to-edge interconnection of two sample ASes, AS 12182 (Internap) and AS 4685 (Asahi-Net ISP), that have had between 6 and 12 AS providers in the last years. We chose these two ASes because both of them actively use AS path prepending at different levels with most of their providers, i.e., both perform actively Internet traffic engineering and would take benefit from our framework. We set $A = 50$ in (3.1) to guarantee that forward path costs and locator costs have similar scales.

We used Routeviews [42] routing tables to qualify the AS graph, path prepending, and path diversity between gateways and locators (i.e., Ω). We used 197 successive 3-day spaced routing tables from Jan. 2009 to Aug. 2010, so as to emulate successive game settings (providers, AS paths and path prepending often change, indeed). Datasets and MATLAB codes are available in [43].

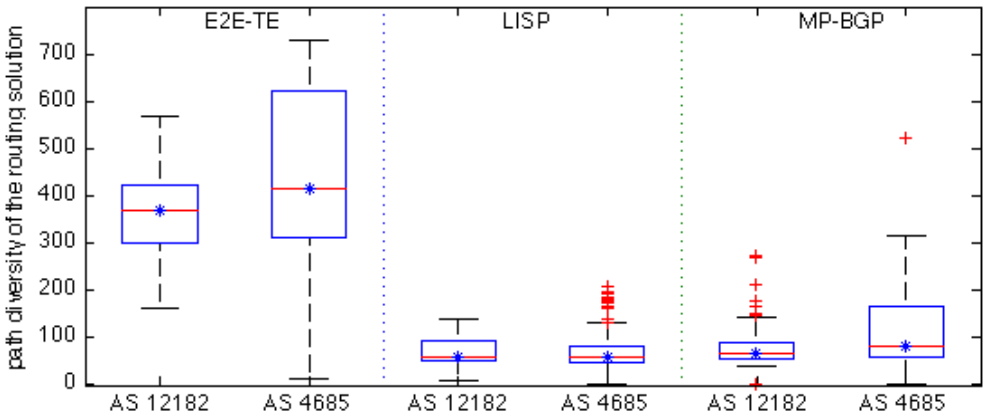


Figure 3.5: Boxplot statistics of path diversity for the solutions.

In the following, we evaluate the performance of our solution (marked 'E2E-TE'). We compare it with the multipath BGP solution ('MP-BGP') and with the solution that one would obtain with normal Locator/ID separation protocol (LISP) [3] (marked with 'LISP'- i.e., the naive case presented in Sect. 3.1), with respect to the routing cost (Fig. 3.4), path diversity (Fig. 3.5) and routing stability(Fig. 3.6) - hence resiliency - of the solution. For the sake of a fair comparison between LISP and MP-BGP, we consider that in LISP a single least-cost locator is chosen and that, if multiple equal-length path are available to the locator, multipath is used among them. We use boxplots to display solution statistical properties (each box, between the min. and the max., displays the first quartile, the median with a '*', third quartile).

Fig. 3.4 depicts the routing cost statistics, showing that while MP-BGP offers an inefficient solution with a cost about twice higher, there are no major differences between our method and the LISP solution based on locator cost minimization. This reflects that our approach does not merely follow the minimization of the routing cost, but is rather more sensible to the strategic pertinence of the routing profile.

Fig. 3.5 shows how many diverse AS-paths are available along the selected gateway-to-locator transit routes, for both routing directions from AS 12182 and AS 4685 (opportunely weighted accordingly to the load sharing distribution, if any), and for the three solution

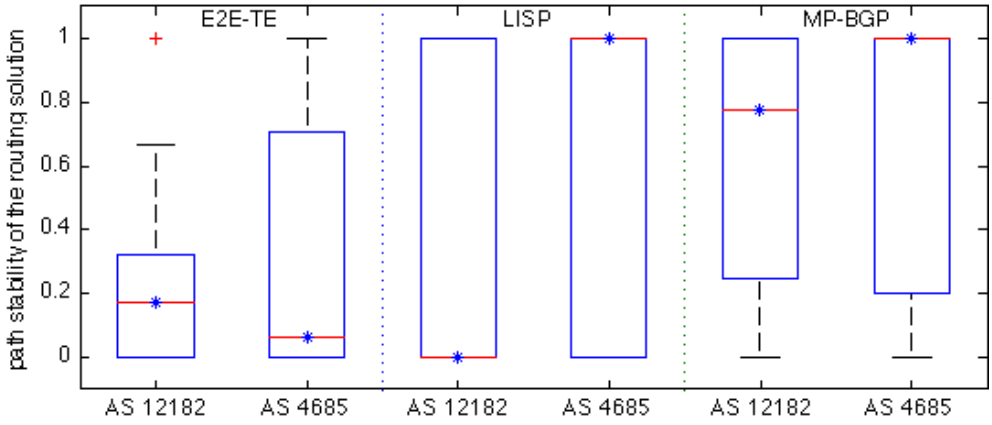


Figure 3.6: Boxplot statistics of routing stability for the solutions.

methods, respectively. While the analysis of routing cost does not show relevant differences, one can appreciate how important improvements can be reached in terms of Internet reliability: we pass from a median of about 50 paths with both MP-BGP and LISP to a median around 400 with our approach[3]. This shows that resiliency route cost functions as intuitive and simple as (3.1) can allow reaching significant improvements with respect to legacy protocols.

Fig. 3.6 shows how much percentage of traffic has been moved at each new solution. The higher it is, the less stable the previous solution can be considered (an instability of 1 indicates that 100% of the traffic volume has be rerouted across different paths). MP-BGP shows a quite high instability, which is in fact not a surprise, with a median above 70%. LISP shows a very high variance and opposite behaviors for the two networks, this probably relates to the fact that AS 12182 reconfigures much more often the path prepending than AS 4685 for traffic engineering purposes. All in all, our method clearly offers a more resilient solution in terms of Internet routing stability with (a median of) less than 10% of the traffic rerouted at each new reconfiguration.

[3]It is worth mentioning that these can be considered too high numbers for real cases; we indeed counted all the loop-free available paths collected exploring Routeviews routing tables; in reality, this number is expected to be much lower due to policy filtering (not visible via routing tables) and limited visibility.

Chapter 4: Generalized Solution to Multipath Routing Using Game Theory

We make an assumption in chapter 3 that the routing cost for each node is the sum of its transit cost and its gateway/locator preference cost, while the preference cost and the transit cost are in the same unit or can be translated into the same unit, which turns the routing game into a potential game and eventually brings us the multipath routing framework to increase the resiliency of Internet routing. The assumption is essential in our previous analysis, but it is not always valid.

When an edge node has several transit nodes to rely on, it is common that the edge node has difference preferences towards different transit nodes for the ingress and egress traffic. The preference can be based on price, security, bandwidth, etc. The paths between transit nodes can be evaluated by certain path evaluation methods, and those methods can take account of the path diversity, delay, path instability, throughput, routing policy, etc. In this dissertation, we define our own path cost function as (3.1) in chapter 3. Our cost function mainly focuses on the path length and path diversity. Although evaluating the preference cost and the transit cost with the same metric is not irrational, the difference between the nature of the two costs determines that we should distinguish them in order to describe more general scenarios.

In this chapter, we generalize that model, accounting for the difference between the nature of *transit cost* and *preference cost* via a vectorized routing cost model, which provides the ability of considering multiple metrics simultaneously.

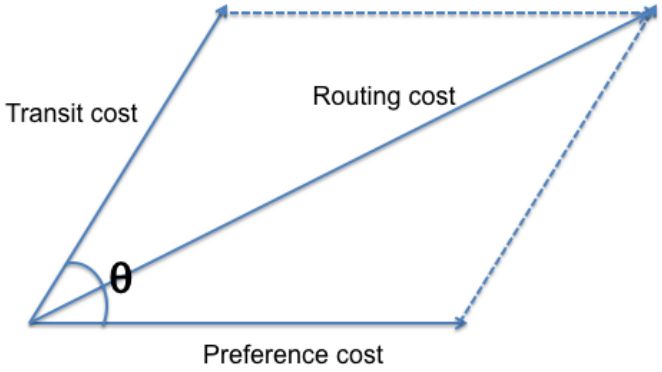


Figure 4.1: An illustration of vectorized routing cost

4.1 A Vectorized Routing Cost Model

Let us assume that transit and preference costs are expressed as vectors, and the routing cost for one node is the sum of its transit cost and the preference cost, while the sum calculation is performed in the vector space. Under such an assumption, it is possible for the two nodes to agree in jointly routing their traffic under certain coordination equilibria, if a proper game theory model can be proposed.

An illustration for the vector sum calculation is shown in Fig. 4.1. In Fig. 4.1, the angle between the vectors of the preference cost and the transit cost is θ. Clearly, the norm of the routing cost is not only decided by the norms of the other two cost vectors, but also affected by the angle θ between the two vectors. The angle θ is set according to the relationship between the nature of the two cost vectors.

Usually edge nodes set the preference costs locally, while the transit cost can be evaluated by certain path evaluation method. Therefore, the two vectors do not always have strong relationship, and is acceptable to assume that they are orthogonal to each other for demonstration purpose. Hence, we set $\theta = \pi/2$ in our analysis. We use real number to denote the preference cost, use imaginary number to denote the routing cost, and use complex number to denote routing cost, which is the sum of reference cost and transit cost.

Hence, it is necessary to re-define the path evaluation function with imaginary number first.

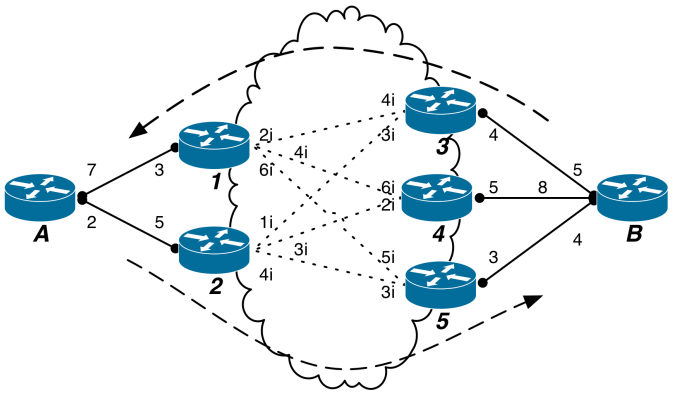


Figure 4.2: Edge-to-edge routing interaction example with vectorized routing costs

Let $\Omega_{i,j}$ be the set of available paths between a gateway i and a locator j, and let $L(\omega)$ be the node hop count of the path $\omega \in \Omega_{i,j}$. We believe it is appropriate to model the set of paths along a node as a system of resistors in parallel, where a resistance corresponds to a path length, and the equivalent resistance (L_{eq}) can be computed. Lengthy paths bring more negligible contributions, and the more available paths the lower route cost. In the following, $c_{i,j}$ represents the transit path costs from the source toward the destination passing by the source's gateway i and destination's locator j, then:

$$c_{i,j} = \lceil A \cdot L_{eq} \rceil \, i = \left\lceil A\left(\sum_{\omega \in \Omega_{i,j}} \frac{1}{L(\omega)}\right)^{-1} \right\rceil i \tag{4.1}$$

where A is an arbitrary constant.

A sample topology is shown in Fig. 4.2, where node i, $i \in \{1, 2, 3, 4, 5\}$ are transit nodes and node A and B are the nodes that exchange a relevant amount of traffic in a stable manner in this large topology network. A knows node 1 and 2 can be used to communicate with B as well as the corresponding costs, and vice versa. The preference cost, shown as a real number, is the cost for edge node A (resp. B) to transmit traffic to transit nodes i and receive traffic from transit nodes i, while the transit cost, shown as an imaginary number, is the cost to transit traffic from i to j. The strategies of players are noted as

G_iL_j, where i and j index the gateway nodes and the locator nodes. For instance, the strategy of G_1L_4 for A suggests to route the flow across gateway 1 toward locator 4 on the way for the destination, which does not only determine that A's transit cost is $6i$ and a part of A's preference cost is 3, but also sets a part of B's preference cost as 8.

4.2 Mathematical Notations

Suppose A's (resp. B's) strategy set is X (resp. Y). Let us distinguish the effect of A's (resp. B's) strategy $x \in X$ (resp. $y \in Y$) to A (resp. B) as $\phi_s(x)$ (resp. $\psi_s(y)$) and to B (resp. A) as $\psi_d(x)$ (resp.$\phi_d(y)$). The routing game can be formalized as $G = (X, Y; f, g)$, where f (resp. g) is the cost function for A (resp. B). Then,

$$\begin{aligned} f(x,y) &= norm(\phi_s(x) + \phi_d(y)) \\ &= \sqrt{(Re(\phi_s(x)) + \phi_d(y))^2 + (Im(\phi_s(x)))^2} \\ g(x,y) &= norm(\psi_d(x) + \psi_s(y)) \\ &= \sqrt{(Re(\psi_s(y)) + \psi_d(x))^2 + (Im(\psi_s(y)))^2} \end{aligned} \tag{4.2}$$

Given $x,\ x' \in X$ and $y,\ y' \in Y$, it is not possible to guarantee that $f(x,y) + f(x',y') - f(x',y) - f(x,y') = g(x,y) + g(x',y') - g(x',y) - g(x,y')$ is always valid based on (4.2). Therefore, the vectorized routing game is not a cardinal potential game.

The non-cooperative game can be formalized as a bimatrix game in Table 4.1. In Table 4.1, the strategies are noted as G_iL_j, where i and j index the gateway nodes and the locator nodes. In each cell of table, the first number is the norm of routing cost for A, the second number is the norm of routing cost for B.

With reductio ad absurdum, it is also possible to prove that the non-cooperative game in

Table 4.1: Vectorized routing game

A \ B	G_3L_1	G_3L_2	G_4L_1	G_4L_2	G_5L_1	G_5L_2
G_1L_3	10.77,9.22	6.40,9.06	10.77,10.77	6.40,10.44	10.77,10	6.40,8.94
G_1L_4	11.66,12.17	7.81,12.04	11.66,13.60	7.81,13.34	11.66,12.53	7.81,11.70
G_1L_5	11.18,8.25	7.07,8.06	11.18,9.85	7.07,9.49	11.18,9.22	7.07,8.06
G_2L_3	12.37,9.22	7.62,9.06	12.37,10.77	7.62,10.44	12.37,10	7.62,8.94
G_2L_4	12.17,12.17	7.28,12.04	12.17,13.60	7.28,13.34	12.17,12.53	7.28,11.70
G_2L_5	12.37,8.25	7.62,8.06	12.37,9.85	7.62,9.49	12.37,9.22	7.62,8.06

Table 4.1 is not a cardinal potential game. Since if we suppose the game is a cardinal potential game and set the corresponding potential values following the rule of cardinal potential game, the potential values cannot satisfy all the conditions for cardinal potential games.

As our previous multipath load sharing framework in chapter 3 is only valid for cardinal potential games, we need to find another tool, other than the potential value, to build the multipath routing equilibrium set.

4.3 An Moderate Approach

The rationale behind NE is that a player would alter his strategy if and only if his current strategy cannot bring him the minimum cost given other players' strategies. A similar concept also works for epsilon-Equilibrium (or near-Nash equilibrium) [44], that is a player would change to the minimum cost strategy given other players' strategies, if and only if the saving is not less than ϵ, a preset threshold.

In the above concept, the players are assumed to be aggressive, as they always compare their current costs with the minimum costs they can achieve given other players' strategies, even though the minimum costs are not stable due to the same behaviors of other players. Meanwhile, all the players would end up achieving the cost in a NE without any coordination. Instead of comparing with the minimum costs that they can achieve given other players' strategies, we propose to set comparison bases in a practical fashion with their costs

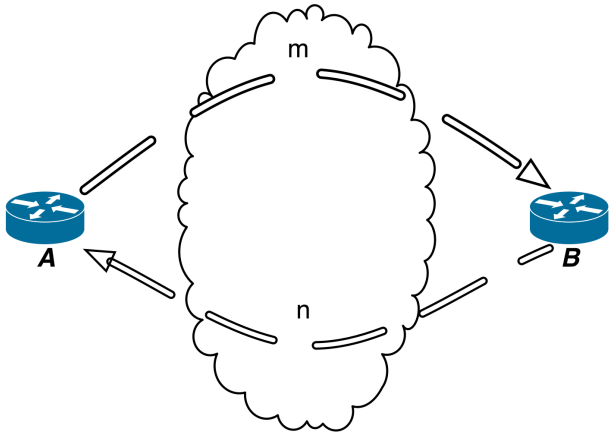


Figure 4.3: A general representation of multipath routing.

in the NE, the costs they should expect without any coordination.

Consider two nodes A and B, as depicted in Fig. 4.3, are connected by m and n paths in the directions A to B and B to A, respectively. Each path is assumed traversing across a multiplicity of hops. For each bi-directional path between nodes A and B (A to B and B to A paths), we consider the associated metric for each node to be a function of costs of the two paths in each direction. The costs can represent the attributes of the path such as traffic load, delay, etc.

For a general bimatrix game, suppose node A has m strategies available, denoted as $S_A = \{s_1, ..., s_m\}$, while node B has n strategies available, denoted as $S_B = \{s_{m+1}, ..., s_{m+n}\}$. Suppose the cost matrix for node A (resp. B) is $\mathbf{U_A}$ (resp. $\mathbf{U_B}$) with dimensions $m \times n$.

For this general routing game, we need to find the expecting routing costs for node A and B.

4.3.1 A Resolution Based on Mixed Strategy

A mixed strategy is a probability distribution vector over all possible strategies, and it represents the probabilities that a player chooses his available pure strategies. A mixed

NE is a point that no player can improve his payoff by unilaterally changing his probability vector over all his possible strategies. For a pair of probability vector $(\vec{p}, \vec{q})$, $\vec{p} = \{p_1, ..., p_m\}^T$ and $\vec{q} = \{q_{m+1}, ..., q_{m+n}\}^T$, it is at NE if and only if for node A either $p_i > 0$ and s_i is the best reply to $\vec{q}$, or $p_i = 0$, while for node B either $q_j > 0$ and s_j is the best reply to $\vec{p}$, or $q_j = 0$.

Let us suppose $m \leqslant n$. At the NE, suppose A applies all of the m strategies as $\vec{p} = \{p_1, ..., p_m\}^T$, then $0 < p_i < 1$ and $\sum_i p_i = 1$. To uniquely find the probability distribution vector $\vec{p}$, we need to have another $m-1$ independent equations. The other $m-1$ independent equations can be achieved from analyzing the behavior of node B. If B choose m of his n available strategies, that means the m strategies have the same cost for B given the action of A, which can provide us $m-1$ independent equations. Therefore, if A applies all of the m strategies, B would choose m of his n strategies. Basing on the m strategies that B would choose, the cost matrix for B is converted into a $m \times m$ matrix, noted as $\mathbf{C_B}$, and $\mathbf{C_B} = [\vec{b}_1, \vec{b}_2, ..., \vec{b}_m]$ where $\vec{b}_i, \forall i \in \{1, 2, ..., m\}$, is a $m \times 1$ column vector.

Suppose the expecting cost for B is v_B, then:

$$\vec{p^T}[\vec{b}_1, \vec{b}_2, ..., \vec{b}_m] = [v_B, v_B, ..., v_B] \tag{4.3}$$

Combining with the initial condition that $\sum_i p_i = 1$, (4.3) can be converted into:

$$\vec{p^T}[\vec{b}_1 - \vec{b}_m, \vec{b}_2 - \vec{b}_m, ..., \vec{b}_{m-1} - \vec{b}_m, \mathbb{1}] = [0, 0, ..., 0, 1] \tag{4.4}$$

where $\mathbb{1}$ is a all one $m \times 1$ column vector that $\mathbb{1} = [1, 1, ..., 1]^T$.

Suppose $\mathbf{K_B} = [\vec{b}_1 - \vec{b}_m, \vec{b}_2 - \vec{b}_m, ..., \vec{b}_{m-1} - \vec{b}_m, \mathbb{1}]$ and $\mathbf{K_B}$ is invertible, then from (4.4) we can get:

$$\vec{p^T} = [0, 0, ..., 0, 1]\mathbf{K_B}^{-1} = [0, 0, ..., 0, 1]\frac{adj(\mathbf{K_B})}{|\mathbf{K_B}|} \tag{4.5}$$

where $adj(\cdot)$ represents the calculation to find out the adjugate matrix of the input matrix.

Suppose $\mathbf{J} = adj(\mathbf{K_B})$, (4.5) can be presented as:

$$\vec{p^T} = [0, 0, ..., 0, 1]\frac{\mathbf{J}}{|\mathbf{K_B}|} = \frac{[\mathbf{J}_{m,1}, \mathbf{J}_{m,2}, ..., \mathbf{J}_{m,m}]}{|\mathbf{K_B}|} \tag{4.6}$$

where $\mathbf{J}_{m,i}$ is the element of matrix J in (m, i).

If we use $\vec{b}_i^{-j}$ to represent a $(m-1) \times 1$ column vector that come from vector $\vec{b}_i$ without the jth element, it is possible to derive that:

$$\begin{aligned} \mathbf{J}_{m,i} &= (-1)^{m+i} \begin{vmatrix} (\vec{b}_1^{-i})^T - (\vec{b}_m^{-i})^T \\ (\vec{b}_2^{-i})^T - (\vec{b}_m^{-i})^T \\ \vdots \\ (\vec{b}_{m-1}^{-i})^T - (\vec{b}_m^{-i})^T \end{vmatrix} \\ &= (-1)^{m+i} |\vec{b}_1^{-i} - \vec{b}_m^{-i}, \vec{b}_2^{-i} - \vec{b}_m^{-i}, ..., \vec{b}_{m-1}^{-i} - \vec{b}_m^{-i}| \\ &= (-1)^{i+m} |\mathbf{M}_{im}| \end{aligned} \tag{4.7}$$

where $\mathbf{M}_{im}$ is the i, m minor matrix of $\mathbf{K_B}$, that is, the $(m-1) \times (m-1)$ matrix from deleting the ith row and the mth column of $\mathbf{K_B}$.

Hence, p_i, $\forall i \in \{1, 2, ..., m\}$, can be calculated as:

$$p_i = \frac{\mathbf{J}_{mi}}{|\mathbf{K_B}|} = (-1)^{i+m}\frac{|\mathbf{M}_{im}|}{|\mathbf{K_B}|} \tag{4.8}$$

Based on Laplace expansion as well as (4.8), v_B can be calculated as:

$$v_B = \vec{p^T} * \vec{b}_m = (-1)^{i+m} \frac{\sum_i |\mathbf{M}_{im}| * b_{im}}{|\mathbf{K_B}|} = \frac{|\mathbf{W_B}|}{|\mathbf{K_B}|} \tag{4.9}$$

where $\mathbf{W_B} = [\vec{b}_1 - \vec{b}_m, \vec{b}_2 - \vec{b}_m, ..., \vec{b}_{m-1} - \vec{b}_m, \vec{b}_m]$.

Since that

$$\mathbf{W_B} = [\vec{b}_1, \vec{b}_2, ..., \vec{b}_m] \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ -1 & -1 & \cdots & -1 & 1 \end{pmatrix} \tag{4.10}$$

Then, we know that $|\mathbf{W_B}| = |\mathbf{C_B}|$, so:

$$v_B = \frac{|\mathbf{C_B}|}{|\mathbf{K_B}|} \tag{4.11}$$

(4.8) shows that a qualified set of strategies for B should satisfy the conditions that $(-1)^{i+m}|\mathbf{M}_{im}|/|\mathbf{K_B}| > 0$, $\forall i \in \{1, 2, ..., m\}$, while (4.11) shows that the strategies set that B should choose is the set that can achieve the minimum value of $|\mathbf{C_B}|/|\mathbf{K_B}|$

After finding which m strategies that B should choose, v_A can be calculated easily. Then v_A and v_B can be used towards the refinement analysis as the comparison bases.

4.3.2 A Resolution Based on the Lemke-Howson Algorithm

The above analysis is based on the assumption that A would apply all of his available strategies. If we loose this constraint, the comparison bases can be obtained with the

Lemke-Howson Algorithm [45].

For a pair of probability vector $(\vec{p}, \vec{q})$, $\vec{p} = \{p_1, ..., p_m\}^T$ and $\vec{q} = \{q_{m+1}, ..., q_{m+n}\}^T$, it is at NE if and only if for node A either $p_i > 0$ and s_i is the best reply to $\vec{q}$, or $p_i = 0$, while for node B either $q_j > 0$ and s_j is the best reply to $\vec{p}$, or $q_j = 0$. Then,

$$\begin{aligned} \mathbf{U_A} \cdot \vec{q} - \vec{r} &= v_A \cdot \mathbb{1} \\ \mathbf{U_B}^T \cdot \vec{p} - \vec{t} &= v_B \cdot \mathbb{1} \end{aligned} \tag{4.12}$$

where $\vec{r} = \{r_1, ..., r_m\}^T$, $\vec{t} = \{t_{m+1}, ..., t_{m+n}\}^T$ and $r_i \geqslant 0$, $t_j \geqslant 0$, while $\mathbb{1}$ indicates a column vector of 1's of appropriate dimension and v_i, $i \in \{A, B\}$ is a scalar representing node i's cost.

Let us use $\mathbb{M}$ to indicate a matrix with the same dimension as $\mathbf{U_A}$ and $\mathbf{U_B}$, and assume all the elements in $\mathbb{M}$ are identical and equal to M, a relatively large number. Then $(\mathbb{M} - \mathbf{U_A}) \cdot \vec{q} + \vec{r} = M \cdot \mathbb{1} - \mathbf{U_A} \cdot \vec{q} + \vec{r} = (M - v_A)\mathbb{1}$. In the same way, $(\mathbb{M} - \mathbf{U_B}^T) \cdot \vec{p} + \vec{t} = (M - v_B)\mathbb{1}$. Suppose $\mathbf{U_A}' = \mathbb{M} - \mathbf{U_A}$, $\mathbf{U_B}' = \mathbb{M} - \mathbf{U_B}$, $v'_A = M - v_A$ and $v'_B = M - v_B$, then we can get (4.13).

$$\begin{aligned} \mathbf{U_A}' \cdot \vec{q} + \vec{r} &= v'_A \cdot \mathbb{1} \\ \mathbf{U_B}'^T \cdot \vec{p} + \vec{t} &= v'_B \cdot \mathbb{1} \end{aligned} \tag{4.13}$$

where all the elements in $\mathbf{U_A}'$ and $\mathbf{U_B}'$ are positive.

The Lemke-Howson algorithm can be applied to find at least one set of practical solutions[1] (p, q) for (4.13). The detailed explanation as well as our implementation for the algorithm is available in our website [43]. Then A and B can ordinate their routing strategies with $(\vec{p}, \vec{q})$ at costs v_A and v_B, respectively. In other words, v_A and v_B are the costs that A and

[1]When the game represented by $\mathbf{U_A}'$ and $\mathbf{U_B}'$ is non-degenerate, the Lemke-Howson algorithm is still capable to find one set of solutions when lexicongraphic method is applied [46].

B should expect.

The problem of finding a NE for finite non-cooperative games has been studied previously by a number of researchers. In our development, we use the Lemke-Howson algorithm [45] for finding a NE, which among the combinatorial algorithms is known to be efficient in practice [47]. The algorithm was first introduced in [46], and was interpreted geometrically in [48]. It visualizes the process of finding NE for when players' strategy set sizes are quite limited. The detailed explanation is in chapter D.

4.3.3 A Linear Load Sharing Approach Based on the Moderate Refinement Method

Since the Lemke-Howson algorithm is more general in application aspect, we decide to use the v_A and v_B got from the Lemke-Howson algorithm as the comparison bases for the moderate refinement method.

Since we find the comparison bases for A and B, we are able to know how dissatisfied A and B are when they are facing other profiles that are not NE. Besides, the stability of one profile is not decided by the players who are satisfied with the profile, but decided by the players who are not satisfied with it. So the maximum value of the players' dissatisfaction towards one profile is a reasonable refinement tool to evaluate the profile.

Therefore, the refinement matrix W is defined as follows:

$$\mathbf{W} = \max(\mathbf{U_A} - \mathbb{V}_A, \mathbf{U_B} - \mathbb{V}_B) \tag{4.14}$$

where the $\max(\cdot, \cdot)$ operation is proceeded separately for each element, matrix $\mathbb{V}_A$ (resp. $\mathbb{V}_B$) has the same dimension as matrix $\mathbf{U_A}$ and $\mathbf{U_B}$ and all the elements of matrix $\mathbb{V}_A$ (resp. $\mathbb{V}_B$) are identical and equal to v_A (resp. v_B).

In this way, when matrix W is used to evaluate each profile's performance, both nodes'

benefit can be considered.

With $\mathbf{W}$, we can include certain profiles, whose refinement value is equal or below a threshold, to achieve better system performance. Here, the refinement value has a similar meaning with ϵ for epsilon-Equilibrium.

Since the maximum and the minimum refinement values change with the game configuration, the threshold can be set accounting for the statistical refinement distribution. An acceptable threshold corresponds to its first quartile.

A further optional step, which is rationally acceptable, is to restrict the solution set only to those that are not Pareto-inferior to any other selected strategies.

Next, we use the refinement values of the remaining solutions to calculate the load sharing distribution ratios, so that a lower refinement value brings to a higher load ratio.

Let $\chi \in S_A \times S_B$ be the set of the profiles kept as solutions, τ be the threshold and $\mathbf{W}_{x,y}$ be the refinement value of $(x, y) \in \chi$. Then,

$$\begin{aligned} \Lambda &= \sum_{(x,y)\in\chi} [1 + \tau - \mathbf{W}_{x,y}] \\ \Lambda_{\widetilde{x}} &= \sum_{(x,y)\in\chi | x=\widetilde{x}} [1 + \tau - \mathbf{W}_{x,y}] \\ \Lambda_{\widetilde{y}} &= \sum_{(x,y)\in\chi | y=\widetilde{y}} [1 + \tau - \mathbf{W}_{x,y}] \end{aligned} \tag{4.15}$$

where Λ represents the total weights, $\Lambda_{\widetilde{x}}$ (resp. $\Lambda_{\widetilde{y}}$) represents the weights for $s_{\widetilde{x}} \in S_A$ (resp. $s_{\widetilde{y}} \in S_Y$).

The traffic is assigned as:

$$\begin{aligned} p_{\widetilde{x}} &= \Lambda_{\widetilde{x}}/\Lambda, \quad \forall s_{\widetilde{x}} \in S_A \\ q_{\widetilde{y}} &= \Lambda_{\widetilde{y}}/\Lambda, \quad \forall s_{\widetilde{y}} \in S_B \end{aligned} \tag{4.16}$$

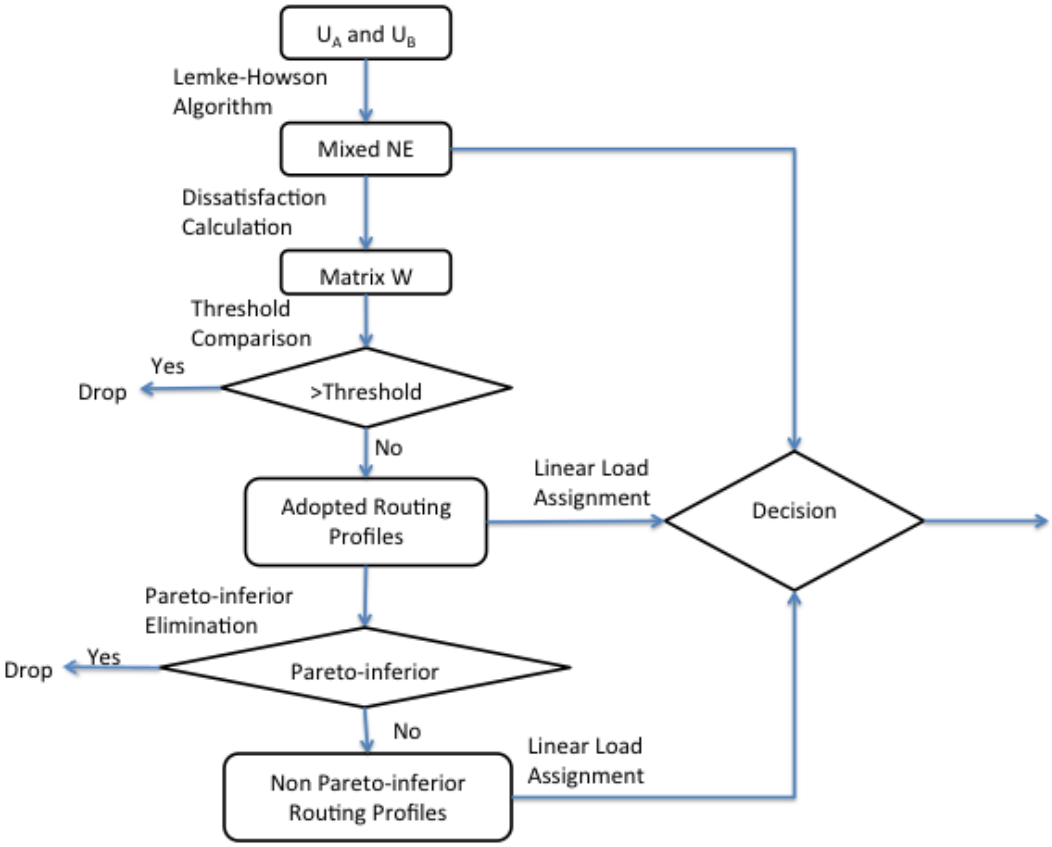


Figure 4.4: The moderate multipath load sharing approach.

We can in this way fairly assign higher weights to those profiles that bear better system performance.

Our approach for applying multipath load sharing is formalized in Fig. 4.4. Through a particular routing cost model, $\mathbf{U_A}$ and $\mathbf{U_B}$ can be calculated. Then, we can use the Lemke-Howson algorithm to find a NE, which is a potential load sharing solution. Through the NE we find, we can continue with the system refinement analysis, and select those profiles with lower refinement values. With those profiles, another potential load sharing solution can be achieved through a linear load assignment basing on their refinement values. The previous selected profiles can be further filtered to exclude Pareto-inferior profiles, and the third potential load sharing solution can be obtained from the left profiles. For a practical multipath load sharing framework, we also need to define a decision criterion to make

Table 4.2: Vectorized routing game with moderate refinement value

A \ B	G_3L_1	G_3L_2	G_4L_1	G_4L_2	G_5L_1	G_5L_2
G_1L_3	(10.77,9.22)$^{4.37}$	**(6.40,9.06)**$^{0.11}$	(10.77,10.77)$^{4.37}$	(6.40,10.44)$^{1.50}$	(10.77,10)$^{4.37}$	**(6.40,8.94)**^{0}
G_1L_4	(11.66,12.17)$^{5.26}$	(7.81,12.04)$^{3.10}$	(11.66,13.60)$^{5.26}$	(7.81,13.34)$^{4.40}$	(11.66,12.53)$^{5.26}$	(7.81,11.70)$^{2.76}$
G_1L_5	(11.18,8.25)$^{4.78}$	**(7.07,8.06)**$^{0.67}$	(11.18,9.85)$^{4.78}$	**(7.07,9.49)**$^{0.67}$	(11.18,9.22)$^{4.78}$	**(7.07,8.06)**$^{0.67}$
G_2L_3	(12.37,9.22)$^{5.97}$	**(7.62,9.06)**$^{1.21}$	(12.37,10.77)$^{5.97}$	(7.62,10.44)$^{1.50}$	(12.37,10)$^{5.97}$	**(7.62,8.94)**$^{1.21}$
G_2L_4	(12.17,12.17)$^{5.76}$	(7.28,12.04)$^{3.10}$	(12.17,13.60)$^{5.76}$	(7.28,13.34)$^{4.40}$	(12.17,12.53)$^{5.76}$	(7.28,11.70)$^{2.76}$
G_2L_5	(12.37,8.25)$^{5.97}$	**(7.62,8.06)**$^{1.21}$	(12.37,9.85)$^{5.97}$	**(7.62,9.49)**$^{1.21}$	(12.37,9.22)$^{5.97}$	**(7.62,8.06)**$^{1.21}$

the final decision among the three solutions, i.e., punishing single path solution to enforce multipath load sharing.

Go back to the Fig. 4.2, with the moderation refinement tool, Table 4.2 can be achieved. In each cell of table, the first number is the norm of routing cost for A, the second number is the norm of routing cost for B, and the exponent is the corresponding moderate refinement value.

In Table 4.2, the first quartile of the refinement values is equal to 1.21. Hence, the routing solution set includes 10 routing profiles, which are bold and have refinement values from 0 to 1.21 in Table 4.1. The threshold actually can be adjusted according to the needs, and more conservative threshold levels other than the first quartile can be used for very large circumstances. After the Pareto-inferior consideration, only 3 profiles are left, which are underlined in the table. We can in this way fairly assign higher weights to those profiles that bear better system performance. For example, in Table 4.1, we obtain the load sharing solution as $p_{G_1L_3} = 42\%$ and $p_{G_1L_5} = 58\%$ for A, while $q_{G_3L_2} = 71\%$ and $q_{G_5L_2} = 29\%$ for B.

4.4 An Aggressive Approach

Corresponding the moderate approach that each node treat the expecting cost as comparison basis, an aggressive approach can be obtained in the sense that the nodes always compare their current costs with the minimum costs they can achieve given other nodes' strategies.

4.4.1 A Resolution Based on Minimum Costs

Aggressive players always compare their current costs with the minimum costs they can achieve given other players' strategies. Therefore, the minimum costs that they can achieve given other players' strategies indeed can also be their comparison bases. In this sense, it is possible to know how dissatisfied the players are when they are facing each profile. So the sum value of the players' dissatisfaction towards one profile is a reasonable refinement tool to evaluate the profile.

For our vectorized routing model, the comparison bases for node A and B are denoted as vectors $\vec{r}$ and $\vec{c}$, then

$$\begin{aligned} \vec{r} &= \min(\mathbf{U_A}) \\ \vec{c} &= \min(\mathbf{U_B^T}) \end{aligned} \tag{4.17}$$

where matrix $\mathbf{U}_A$ (resp. $\mathbf{U}_B$) is the cost matrix for A (resp. B) and the $\min(\cdot)$ operation treats the columns of the input matrix as vectors, returning a row vector containing the minimum element from each column.

In order to considered both nodes' benefits during coordination, we define the equilibrium refinement tool matrix $\mathbf{W}$ as follows:

$$\mathbf{W}_{i,j} = (\mathbf{U_A}_{i,j} - r_j) + (\mathbf{U_B}_{i,j} - c_i) \tag{4.18}$$

where $\mathbf{W}_{i,j}$ (resp. $\mathbf{U_A}_{i,j}$ and $\mathbf{U_B}_{i,j}$) is the (i,j) element in matrix $\mathbf{W}$ (resp. $\mathbf{U_A}$ and $\mathbf{U_B}$), while r_j (resp. c_i) is the jth (resp. ith) element in vector $\vec{r}$ (resp. $\vec{c}$).

For the scalar routing model in [7], we showed that the routing game could be decomposed

into a selfish game and a dummy game, so:

$$\begin{aligned} \mathbf{A}_{i,j} &= \phi_s(i) + \phi_d(j) \\ \mathbf{B}_{i,j} &= \psi_d(i) + \psi_s(j) \end{aligned} \tag{4.19}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the cost matrices for node A and B in the scalar model.

Following (4.17), the comparison bases for the scalar model is:

$$\begin{aligned} r_j &= \phi_s(\widetilde{i}) + \phi_d(j) \\ c_i &= \psi_d(i) + \psi_s(\widetilde{j}) \end{aligned} \tag{4.20}$$

where $\phi_s(i)$ and $\psi_s(j))$ achieve the minimum value at $\widetilde{i}$ and $\widetilde{j}$, respectively.

Then, if we carry on the universal refinement analysis proposed above, it can be shown that:

$$\begin{aligned} \mathbf{Z}_{i,j} - \mathbf{Z}_{i',j} &= \mathbf{A}_{i,j} - \mathbf{A}_{i',j} \\ \mathbf{Z}_{i,j} - \mathbf{Z}_{i,j'} &= \mathbf{B}_{i,j} - \mathbf{B}_{i,j'} \end{aligned} \tag{4.21}$$

where $\mathbf{Z}$ is the refinement matrix for the scalar routing model.

From (4.21), we find that the aggressive refinement values become potential values when the vectorized game is degenerated into a potential game. Therefore, if we treat the vectorized routing game as an extension of the scalar game, the aggressive refinement method can be seen as a generalized method of the potential value evaluation.

With $\mathbf{W}$, we can select the routing profiles, whose refinement value is equal or below a threshold, to achieve better system performance. Here, the refinement value the same meaning with ϵ for epsilon-Equilibrium.

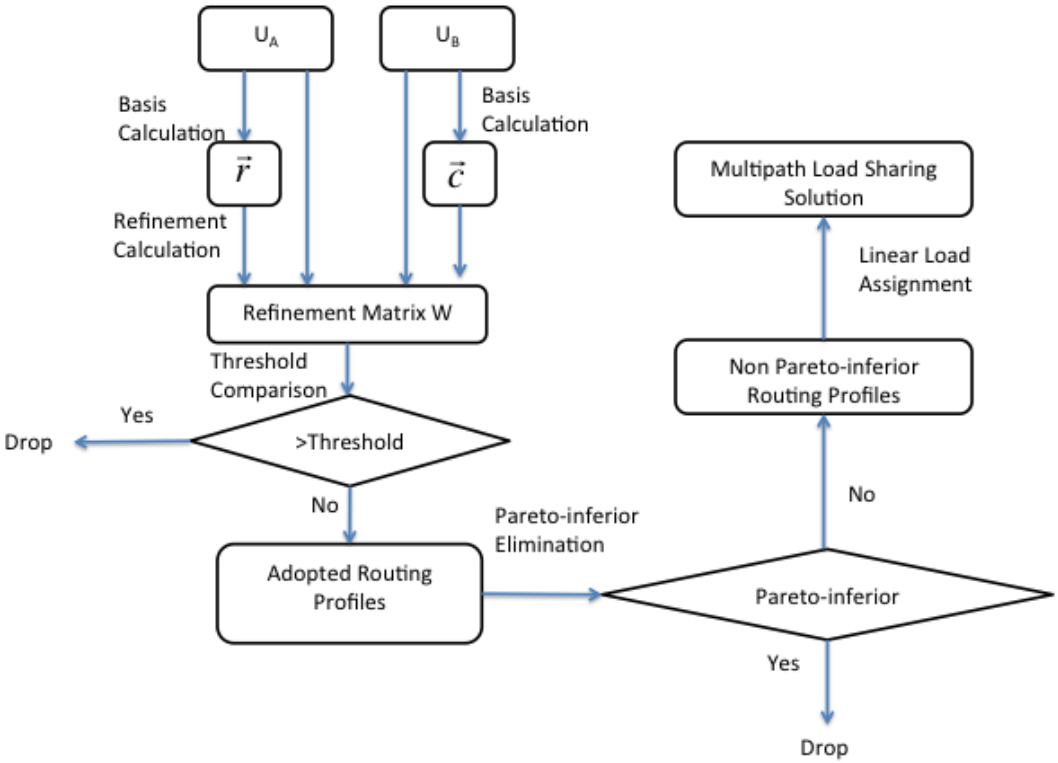


Figure 4.5: The aggressive multipath load sharing approach.

Table 4.3: Vectorized routing game

A \ B	G_3L_1	G_3L_2	G_4L_1	G_4L_2	G_5L_1	G_5L_2	
G_1L_3	**(10.77, 9.22)**$^{0.28}$	**(6.40, 9.06)**$^{0.11}$	(10.77, 10.77)$^{1.83}$	(6.40, 10.44)$^{1.50}$	**(10.77, 10)**$^{1.06}$	**(6.40, 8.94)**^{0}	8.94
G_1L_4	(11.66, 12.17)$^{1.35}$	(7.81, 12.04)$^{1.74}$	(11.66, 13.60)$^{2.79}$	(7.81, 13.34)$^{3.04}$	(11.66, 12.53)$^{1.72}$	(7.81, 11.70)$^{1.41}$	11.70
G_1L_5	**(11.18, 8.25)**$^{0.59}$	**(7.07, 8.06)**$^{0.67}$	(11.18, 9.85)$^{2.20}$	(7.07, 9.49)$^{2.09}$	(11.18, 9.22)$^{1.57}$	**(7.07, 8.06)**$^{0.67}$	8.06
G_2L_3	(12.37, 9.22)$^{1.87}$	(7.62, 9.06)$^{1.32}$	(12.37, 10.77)$^{3.43}$	(7.62, 10.44)$^{2.71}$	(12.37, 10)$^{2.65}$	**(7.62, 8.94)**$^{1.21}$	8.94
G_2L_4	(12.17, 12.17)$^{1.86}$	(7.28, 12.04)$^{1.32}$	(12.17, 13.60)$^{3.29}$	(7.28, 13.34)$^{2.51}$	(12.17, 12.53)$^{2.22}$	**(7.28, 11.70)**$^{0.88}$	11.70
G_2L_5	(12.37, 8.25)$^{1.78}$	**(7.62, 8.06)**$^{1.21}$	(12.37, 9.85)$^{3.39}$	(7.62, 9.49)$^{2.63}$	(12.37, 9.22)$^{2.76}$	**(7.62, 8.06)**$^{1.21}$	8.06
	10.77	6.40	10.77	6.40	10.77	6.40	$r\backslash c$

Our approach for applying aggressive multipath load sharing is formalized in Fig. 4.5. Through the vectorized routing cost model, $\mathbf{U_A}$ and $\mathbf{U_B}$ can be calculated. Then, we can use $\mathbf{U_A}$ and $\mathbf{U_B}$ to calculate the refinement matrix and select those profiles with lower refinement values. The previous selected profiles can be further filtered to exclude Pareto-inferior profiles, and the load sharing solution can be obtained from the left profiles. It is worth noting that this framework does not rely on NE existence, which does not need to be guaranteed.

In Table 4.3, the first quartile of the dissatisfaction values is equal to 1.21. Hence, the routing solution set includes 11 routing profiles, which are bold and have refinement values from 0 to 1.21 in Table 4.1. The threshold actually can be adjusted according to the needs, and more conservative threshold levels other than the first quartile can be used for very large circumstances. After the Pareto-inferior consideration, only 3 profiles are left, which are underlined in the table. We can in this way fairly assign higher weights to those profiles that bear better system performance. For example, in Table 4.1, we obtain the load sharing solution as $p_{G_1L_3} = 42\%$ and $p_{G_1L_5} = 58\%$ for A, while $q_{G_3L_2} = 71\%$ and $q_{G_5L_2} = 29\%$ for B.

4.5 Performance Evaluation

We simulated the generalized edge-to-edge routing framework with two sample ASes, AS 12182 (Internap) and AS 4685 (Asahi-Net ISP). We chose these two ASes because both of them actively use AS path prepending at different levels with most of their providers, i.e., both perform actively Internet traffic engineering and would take benefit from our framework.

We used Routeviews [42] routing tables to qualify the AS graph, path prepending, and path diversity between gateways and locators (i.e., Ω). We used 197 successive 3-day spaced routing tables from Jan. 2009 to Aug. 2010, so as to emulate successive game settings (providers, AS paths and path prepending often change, indeed). Datasets and MATLAB codes are available in [43].

In the following, we evaluate our aggressive solution (marked 'GE2E-TE'). First, we characterize the non-Pareto-inferior equilibria set (Fig. 4.6) given by our solution. Then, we compare the performance of our solution with the multipath BGP solution (marked 'MP-BGP') and with the solution that one would obtain with normal Locator/ID separation protocol (LISP) [3] (marked with 'LISP'), with respect to the total routing cost (Fig. 3.4), path diversity (Fig. 3.5) and path stability. For the sake of a fair comparison among LISP

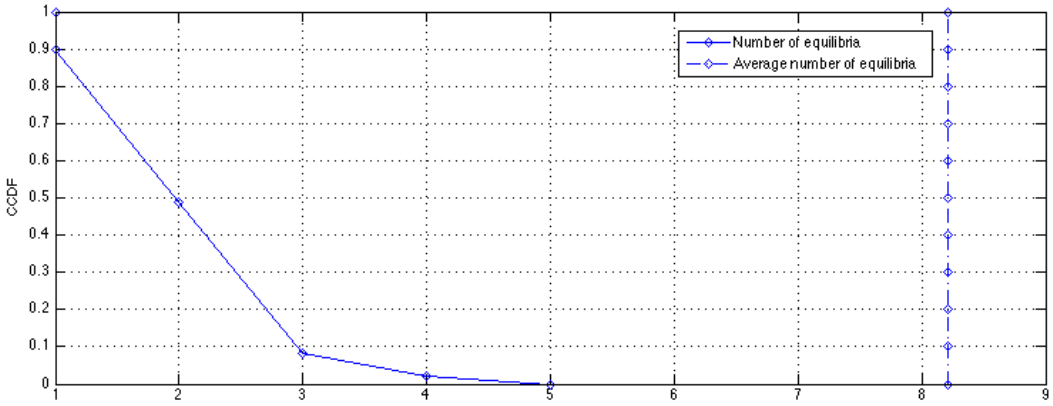


Figure 4.6: The CCDF of the size of non-Pareto-inferior equilibria set.

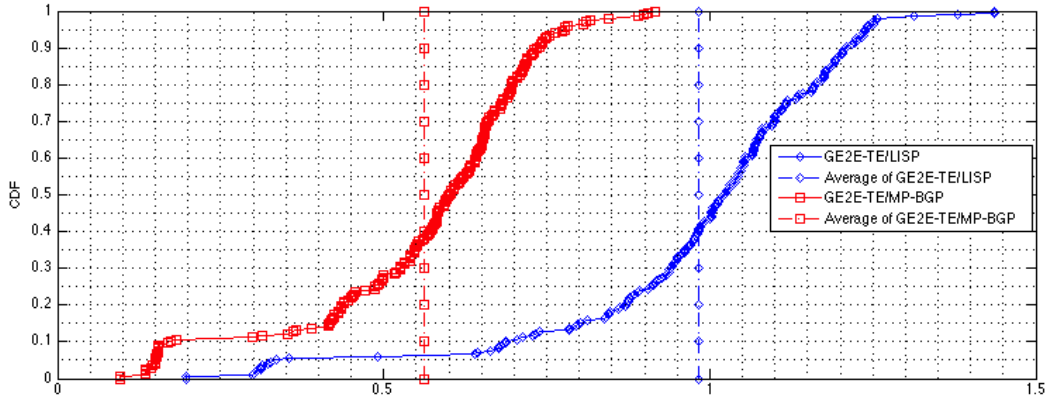


Figure 4.7: The CDFs of routing cost ratio for the solutions.

and MP-BGP, we consider that in LISP a single least-cost locator is chosen and that, if multiple equal-length path are available to the locator, multipath is used among them.

Fig. 4.6 shows the complementary cumulative distribution function (CCDF) and the average of the size of non-Pareto-inferior equilibria set given by our solution. We find that the probability to achieve multiply equilibria is around 0.9, and the average size of the equilibria set is larger than 8. Therefore, our solution framework is eligible in introducing multipath load sharing.

We divide the routing cost of our algorithm with the costs of MP-BGP and LISP, and use cumulative distribution functions (CDFs) to depict the comparison. With MP-BGP,

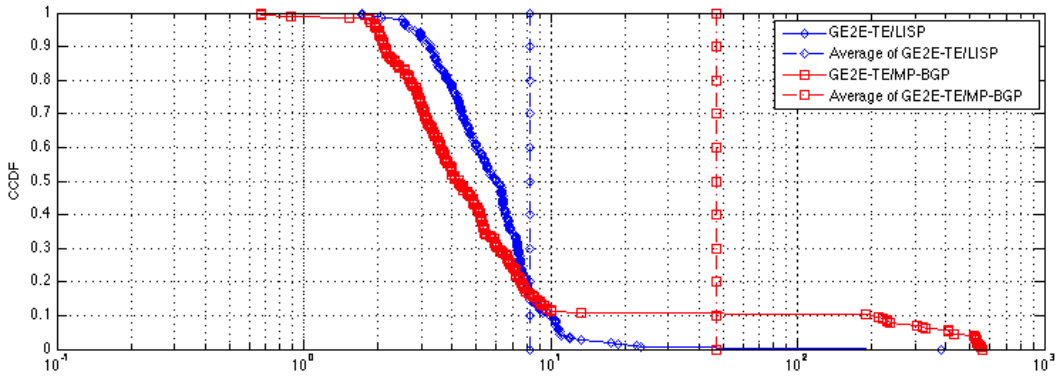


Figure 4.8: The CCDFs of path diversity ratio for the solutions.

Fig. 4.7 shows that our algorithm always works better since the CDF goes to 1 when the ratio is around 0.9. From the average, we can see that the cost of our algorithm is roughly 56% of that of MP-BGP, which means it is possible to save around 44% if MP-BGP adopts our algorithm. With LISP, the probability that our algorithm works better is roughly 0.45, and the average cost of our algorithm is around 98% of that of LISP.

In the simulation, our approach does not outreach LISP significantly in the aspect of routing cost, and the reason is that our approach does not merely chase the routing profiles that can minimize the routing cost, but is rather more sensible to the profiles that can benefit the two players simultaneously. Thus, the solutions our approach brings are beneficial from a bilateral standpoint.

We compare the routing path diversity of our algorithm to that of MP-BGP and LISP, and use CCDFs to depict the comparison. Fig. 4.8 shows that our algorithm outperforms the other protocols in terms of path diversity. The probability that our algorithm can offer a path diversity at least twice as many as MP-BGP and LISP can offer is around 1. From the average of the results, we find that the expected increase of path diversity is around 8 times and 45 times, if MP-BGP and LISP adopt our algorithm. This shows that resiliency-aware routing cost functions as intuitive and simple as (4.1) can reach significant improvements with respect to legacy protocols.

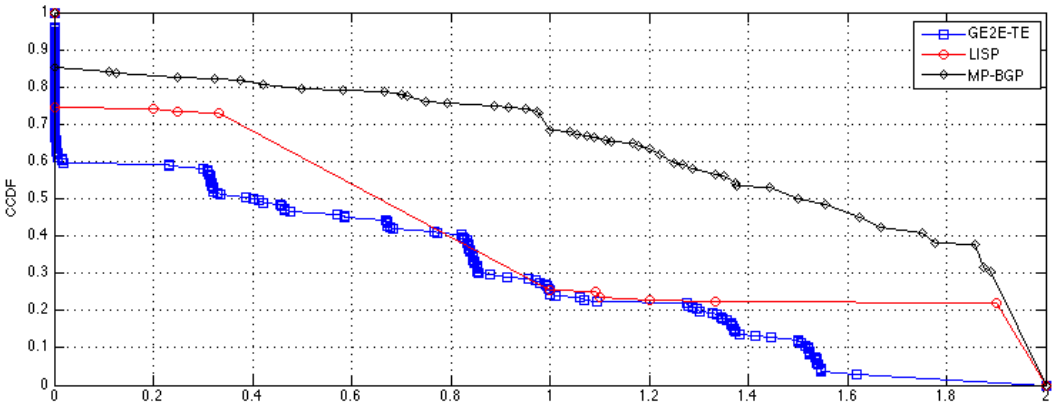


Figure 4.9: CCDFs of routing instability for the solutions.

Fig. 4.9 shows how much percentage of traffic has been moved for both ASes in each consecutive calculation. The higher it is, the less stable the previous solution can be considered (an instability of 1 indicates that 100% of the traffic volume has be rerouted across different paths). The average of path instabilities for GE2E-TE, LISP and MP-BGP are 0.5918, 0.9623 and 1.3055, respectively. In the aspect of path instability, it is shown that our algorithm is almost always better than MP-BGP and LISP. Besides, LISP can also benefit from our algorithm from the average value perspective.

Our method clearly offers a more resilient solution in terms of Internet routing stability. The reason is that the routing profile selection in our method prefers the profiles that can benefit the two parties at the time, and are attractive to both ASes, thus more stable than the other two methods.

4.6 Generalization to N-Networks

We restricted our multipath routing and load sharing framework to a bilateral routing coordination between two edge networks. In this section, we show how it can be easily generalized to more than two networks, and we propose additional traffic engineering enhancements.

4.6.1 Game Extension to Multiple Players

In order to extend the proposed approach into more than two networks, we assume that the traffic exchange among them are significant. In other words, the edge networks with negligible traffic can be considered separately, and will be discarded in the joint modeling.

It is worth noting that if all the edge nodes are to be included in the joint modeling, we would obtain a game with an infinite number of players. Besides being untreatable, this would also be ineffective since we can restrict the game modeling in a pragmatical way, and only include the group of those edge nodes with significant reciprocal traffic volume exchanges. Moreover, such a systematic approach would need to index all the networks, which would be impracticable given the rapid and decentralized evolution of the Internet ecosystem.

4.6.2 Game Strategies, Cluster Size and Complexity Concerns

From a game setting perspective, let us suppose there are N edge nodes that are jointly exchanging traffic, and the game become to a N-player non-cooperative game. Let X_i be the strategy set of the i^{th} edge node, $i \in N$, and let P_i the number of providers/locators of node i. Then, $|X_i| = \Pi^{i \neq j}_{(i,j) \in N \times N}(P_i \cdot P_j)$. For example, in a case of 3 edge nodes with 2, 3 and 4 providers each, respectively, we obtain a set of 48 strategies for node I, 72 strategies for node II and 96 for node III, with a three-dimension array of 331776 elements. A strategy $x \in X_1$ may, e.g., be $x = G_2L_3, G_1L_9$ indicating to route the flow from node I to node II via the gateway 2 and the locator 3 and the flow from node II to node III via the gateway 1 and the locator 9.

Nevertheless, for larger instances with a high number of networks, one may obtain untreatable instances. For the original N edge nodes case, suppose each one has k providers/locators, we obtain sets of k^{2N-1} strategies elements. For large settings (e.g., $k > 5$ and $N > 50$) there may be thus need to define a more scalable and less precise modeling. Very large

instances would be, however, likely uncommon; in any case, a possible technical solution would be to implement multi-cluster settings with per-cluster edge link reservation levels and routing costs (somehow similarly to multi-level topologies for link-state Interior Gateway Protocols).

4.6.3 N Nodes Mathematical Notation

The generalized game is $G = (X_1, ..., X_N; f^1, .., f^N)$, and $f^i = |f^i_s(x_i) + f^i_d(x_{-i})|$ is the cost function of the i^{th} edge node, where x_i is the strategy of node i and x_{-i} is the strategies of other nodes. Note that the cost functions now contain many cost components, one for each flow (whose ingress gateway and egress locator are indicated by the strategy X_i).

For example, if three flows (toward as many destination locators) routed across the same egress edge link, the egress unitary routing cost in f^i_s for that edge link is triplicated. It is worth mentioning that, alternatively to the multiplication of the same link cost by the number of routed flows, in practice there is the opportunity to implement congestion control mechanisms; this can be done by adding congestion cost components to f_s and f_d as function of the used link bandwidth, if flow bandwidths are known by all the networks in the cluster.

Let us denote the cost matrix and comparison basis for node i as U^i and B^i, where U^i is an N dimensional matrix and B^i is an $N-1$ dimensional matrix. Then,

$$B^i_{a_1,...,a_{N-1}} = \min_x(U^i_{a_1,...,a_{i-1},x,a_{i+1},...,a_{N-1}}) \tag{4.22}$$

Denote the refinement matrix as W, where W is also an N dimensional matrix. Then:

$$W_{a_1,...,a_N} = \sum_i (U^i_{a_1,...,a_N} - B^i_{a_1,...,a_{i-1},a_{i+1},...,a_N}) \tag{4.23}$$

Based on the refinement matrix W, it is still possible to select the profiles that bear good system performance. Then a multipath routing and load sharing solution can be achieved with the linear assignment method proposed in previous section.

Chapter 5: Application and Implementation

Certain thoughts towards the application of the proposed traffic engineering framework will be discussed in this chapter. We will first reveal the benefits brought by multi-homing, including the reason that why multi-homing is widely applied, the path diversity brought by multi-homing as well as why the proposed approach can be applied in current Internet. Then we discuss the connection of the proposed framework with Entropy Labels and Locator/Identifier Separation Protocol (LISP).

5.1 The Benefits From Multi-homing

The resiliency of the Internet represents the networks availability, and it plays a vital role within the ecosystem of the Internet, especially in times of instabilities or under extreme conditions. Due to the concerns towards the Internet resiliency issues, the application of multi-homing among destination ASes are very popular.

In Fig. 5.1, we notice that the multi-homing utilization behavior is related to the functionality of the destination ASes. Most service and content destination ASes tend to apply multi-homing, since resiliency is indeed a key factor for their functionality.

Basing on the popularity of multi-homing behavior, the generalized multipath load sharing framework, developed in this dissertation, can be smoothly applied to increase the Internet resiliency. The destination ASes, which have the incentive to improve their routing resiliency, can be seen as edge nodes, while the multi-homing upstreaming ASes can be treated as transit nodes.

This load sharing framework can be applied per packet, however this practice may result

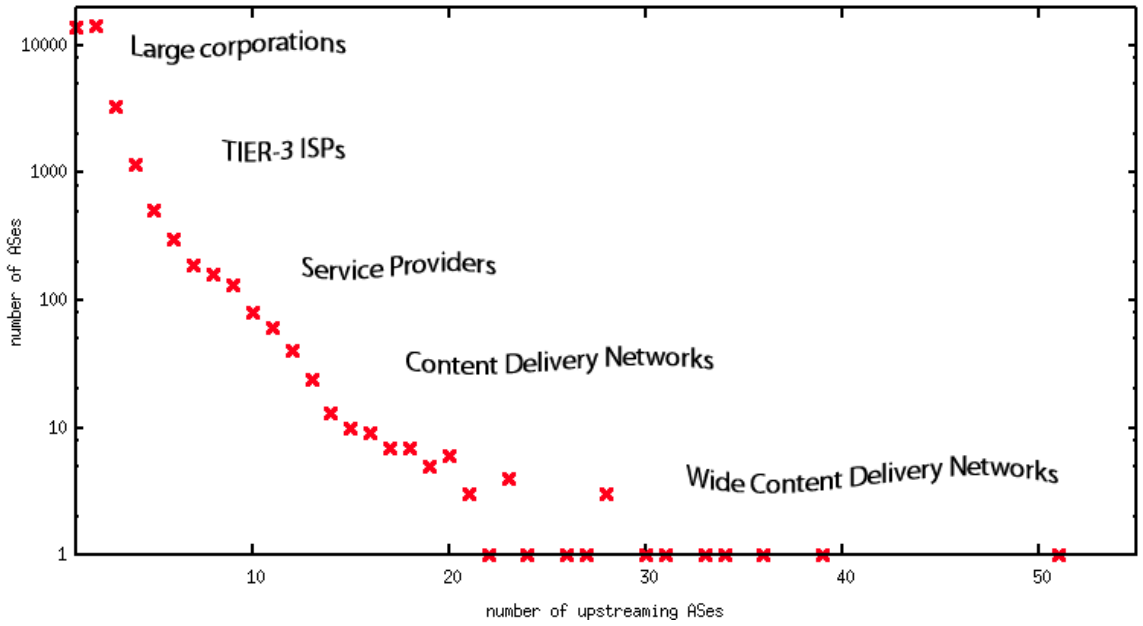


Figure 5.1: Multi-homing distribution of destination ASes (as of 25 Aug., 2010).

in Jitter or delay and even Out-of-Order packets to ultimate destination. To void those issues, the load sharing should be flow specific, especially for Label Switched Paths (LSPs) within Multi-protocol Label Switching (MPLS) enabled networks. Among the available techniques, Entropy Labels, which is under the process of standardization by IETF, deems a practical solution to provide flow specific load sharing.

5.2 Entropy Labels

The concept of Entropy Labels was introduced in the context of "fat pseudowires". Entropy Labels have since been generalized to many more MPLS applications.

The standard of Entropy Labels was first proposed in 2008 to eliminate the need for transit LSRs to perform deep packet inspection for multipath flow routing. After several years evolving, some significant design changes were made, and major Internet vendors have already accepted the idea or similar designs and joined in the implementation, e.g., Juniper, Level 3 Communications, Alcatel-Lucent, Cisco and Huawei [49] [50].

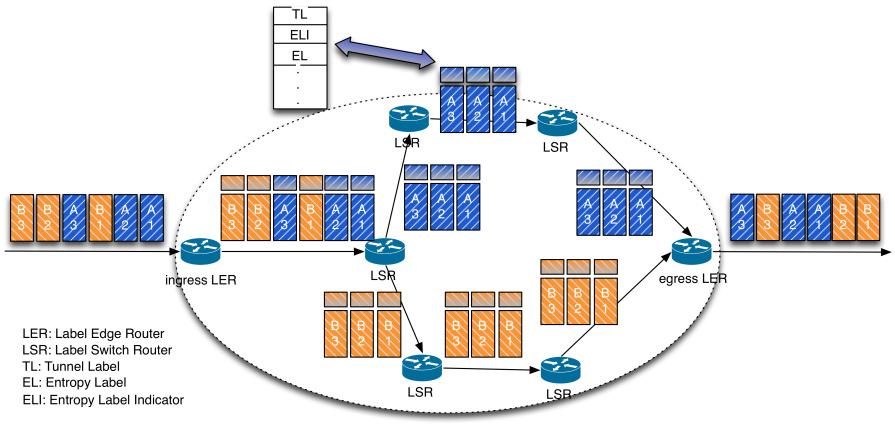


Figure 5.2: An illustration about Entropy Labels

Fig. 5.2 illustrates a simple routing example using Entropy Labels within a MPLS enabled network. The ingress LER pushes labels to the flow packets it receives in accordance with the flow packets characteristics, e.g., Transport protocol (UDP or TCP), source and destination IP address as well as their port numbers. Tunnel labels, usually assigned on the basis of destination IP address as well as certain constrained requirements, control the LSP and different flows (flow A and B in Fig. 5.2) may get the same tunnel labels. When collectively considering the flow packets characteristics as KEYS and inputting the KEYS into a hash function to generate Entropy Labels, it is possible to apply loading sharing routing strategies to different flows with the same tunnel labels.

In Fig. 5.2, flow A and B are routed with different LSPs even they may share the same tunnel labels. In addition, although Entropy Labels cannot guarantee the overall packets sequence the egress LER, the order of packets within the same flow can be insured, as the same flow will get the same LSP.

Entropy Labels technically guarantee the efficiency for multipath routing, but does not provide a concrete algorithm for how to distribute the traffic among the available routes. In this sense, our generalized multipath load sharing framework has the potential to become a solid complementary to it.

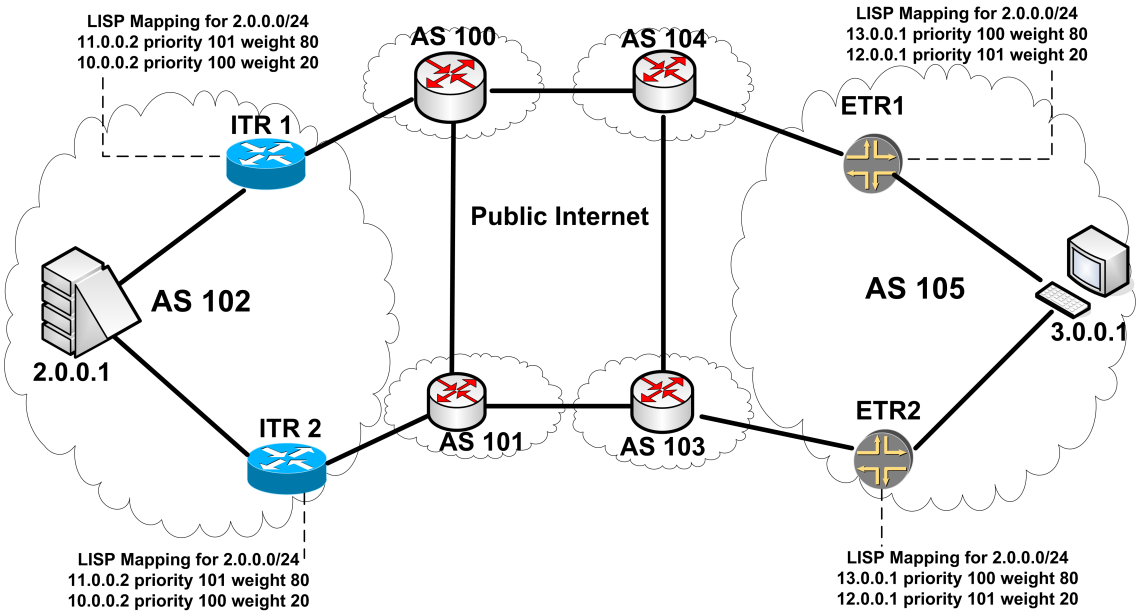


Figure 5.3: Network-based Locator/Identifier Separation study case example

5.3 Locator/Identifier Separation Protocol (LISP)

In network-based solutions, as depicted in Fig. 5.3, a network gateway (called Ingress Tunnel Router, ITR) receives packets with source and destination identifiers (e.g., from 2.0.0.1 to 3.0.0.1), then handles the identifier to locator mapping, encapsulates the packet toward the locator gateway (called Egress Tunnel Router, ETR), which will finally decapsulate the packet and send it to the destination. Network gateways somehow separate edge networks (e.g., AS 102 and AS 105) from the Internet core, and might be managed by core carrier providers without the involvement of edge ASes and customers (i.e., in Fig. 5.3 by functionally moving ITRs and ETRs into the provider network).

Multi-homing in network-based LISP can be managed by allowing the LISP network domain at the edges to announce routing preferences on the incoming traffic. In the current LISP protocol under standardization, priorities and load-balancing weights can be assigned to routing locators so as to give a criterion to the source gateways to choose among many locators (as indicated, e.g., in Fig. 5.3). Using these weights and priorities the edge ASes

can perform inbound traffic engineering, which is not possible with BGP (at least, some BGP attributes allow some forms of inbound traffic engineering that can, however, be nullified by external policies).

Thanks to the new capabilities of edge-to-edge traffic engineering, using the ingress priority and load balancing (weight) features, the connection resiliency is increased. On the other side, the core path diversity still depends on BGP (as far as only BGP is used in the core). Moreover, fixing gateways create node-related resiliency concerns; core traffic engineering methods are needed for fast-rerouting and protection against gateway outages.

The current LISP specification is an IRTF/IETF proposition encompassing transit-edge hierarchical routing, which has been already implemented in some new routers (e.g., in Cisco 7200 routers). In LISP, the translation from destination identifier to routing locators (RLOC) is performed by a distributed database system called mapping system. The mapping systems provides all the locators announced for an identifier or an identifier space, and can optionally set for each locator a priority cost (lower is preferred) and a load sharing integer weight (from 0 to 100) announced by the corresponding network gateways. In its current form, the LISP weight is used only when there are equal LISP priorities. As already argued, the naive usage of such weights and priorities is not strategically justified when the communicating networks are independent and have significant equivalent traffic exchanges. Our framework can be thus seen as a Traffic Engineering LISP (LISP-TE) framework.

From a practical standpoint, we are interested in using integer percentage values out of b_x and b_y (or b_{x_i} for the n/networks case) ratios for backward compatibility with the LISP's integer weights. The LISP priority field might be used as a coordination channel, and might possible be extended to allow the coding of both backward locator cost and forward path costs. It is worth mentioning that the LISP priorities and weights are to be announced globally, while in the bilateral interaction case a private bilateral signaling is needed. For the bilateral case, another coordination channel may be managed independently of, but coupled with, the global LISP mapping system. For the case of a cluster of edge networks,

the load sharing solution obtained can either be similarly restricted to the routing among cluster members only, or can be applied to *any* source edge network if the sum of all the traffic contributions from all other edge networks is negligible with respect to the intra-cluster volume. Therefore, this last setting would be directly implementable under the current LISP proposition.

From an operational standpoint, an appropriate execution policy can be:

- when the LISP priorities are different, extract from them the locator costs and the forward path costs;
- compute the coordinated load sharing solution;
- set the LISP weights accordingly;
- set the LISP priorities equal to each other.

When a network needs to announce new cost settings to reflect changes in traffic characteristics, Internet paths and their performance, or topology properties, it simply resets accordingly the LISP priorities so that the upstream networks detect the change; then, all the participating networks implicitly converge to a new coordinated load sharing solution.

Chapter 6: Internet Hierarchical Interconnection Measurement[1]

In this chapter, we measure the Internet topology from a Transit-Edge (T-E) hierarchical routing perspective. By analyzing the recent BGP tables over a three years and a half period, we aim at characterizing the properties of edge and transit networks from interconnection, routing and traffic engineering perspectives. Those properties are important when considering the application of our algorithm.

We first analyze the diameters and the shortest paths between ASes pairs as well as the degrees and the betweenness of edge and transit ASes. Then we characterize some properties of T-E routing. We also analyze the AS path prepending and IP prefix de-aggregation related properties for edge and transit ASes. Sparked by the definition of betweenness, we quantity the routing importance of edge and transit ASes with a novel metric we define to measure routing centrality. Last but not least, we also define a novel metric to measure the routing instability phenomenon, for edge and transit AS networks.

6.1 Background

At the inception of the Internet, many technology choices had to be taken, such as the forwarding nature of the Internet Protocol, its addressing and the inter-domain routing principle. The history tells us that the Internet Protocol (IP) relies on packet switching with statistical multiplexing, that its addressing is based on a 32-bit space and is now migrating to a 128-bit space, and that the Border Gateway Protocol (BGP) [1] is the inter-domain routing protocol used by ASes to exchange routing information. BGP relies on a

[1] A preliminary version of analysis has been presented at the 2011 Network of the Future Conference (NoF 2011). [2]

flat routing mode using path vectors for each IP network prefix, announced independently in an uncoordinated fashion.

The lack of coordination amongst AS networks appears strategically reasonable, as each AS needs to fulfill its own interests and objectives first. However, the flat routing mode of Internet routing is unable to scale with such a behavior for a very large number of networks. Meanwhile, the number of ASes as well as the announced network prefixes are increasing extremely fast (currently, about 41,000 ASes and 400,000 network prefixes). Such a large and increasing number of prefixes, even if dictated by reasonable traffic engineering and multi-homing practices, are posing many issues from a network management viewpoint. Coupled with other aspects such as BGP routing convergence, instability and weak resiliency, they are undermining the healthy development of the Internet.

The scalability and resiliency issues of the Internet are very active research areas, and numerous research efforts have been devoted to the analysis the factors causing the issues, including the increase rate of BGP updates as well as the BGP routing table size growth [51], [52]. In [53] and [54], the authors mainly focused on the former aspect. They identified the most significant sources of BGP updates and described the contribution of different factors towards the increase rate of BGP updates as the Internet growth. Our work targets the latter aspect. We expect that we can find certain inner connections between the evolution of the Internet and growth of BGP routing table sizes.

A direction recently evaluated to tackle the Internet routing scalability and resiliency issue is to adopt hierarchical routing schemes [52], [55], [7], a promising direction to improve Internet scalability and resiliency by allowing explicit forwarding through routing locators on the way toward the destination network. Allowing a two-level hierarchy routing between edge and transit networks, it is possible to reduce the transit routing table sizes since a very large majority of the Internet networks are at the edges and do not transit traffic. In addition, it is also possible to achieve better user mobility and mitigate important routing security issues through applying the hierarchical routing. Moreover, novel traffic engineering

capabilities can be introduced.

6.2 Transit and Edge Networks

The Internet interconnection graph can be partially inferred via BGP routing tables. Route-views' public routing tables [42] aggregate the daily view of multiple backbone routers, which represents a very detailed mirror on the Internet ecosystem evolution. After a rapid analysis, we find that at present around 84% of the total ASes act as pure destination networks, only appearing at the last position of the AS paths. They are commonly considered as "stub ASes"[2]. In practice, some large stub ASes (content providers and delivery networks) functionally fragment their networks into multiple ASes for management reasons, and they may also appear in the penultimate or in the third from last position in AS paths. Nearly 13% additional ASes appear up to the third from last position of BGP AS paths, among which there are certainly also some regional Internet Service Providers (ISPs). The sub-network composed of these 97% ASes can be seen as the edge of the Internet, which given its interconnection behavior has different traffic engineering requirements and routing purposes than transit networks. In fact, the remaining 3% ASes do transit the global Internet traffic as their principal purpose, and they can be treated as the transit part of the Internet. As of our observation, these transit and edge network ratios have been rather stable even though the Internet has significantly grown.

The T-E hierarchical routing paradigm suggests inserting routing locators at the frontier between transit and edge networks. Different protocols can be conceived to manage identifier-to-locator mappings and to encapsulate or aggregating (tunneling) packets in the transit sub-path. One working at layer 3 only is the Locator-Identifier separation protocol (LISP) [3], currently under standardization (it somehow supersedes host-based approaches such as SHIM6 [56] or HIP [57] that appear as less scalable mechanisms). Besides allowing

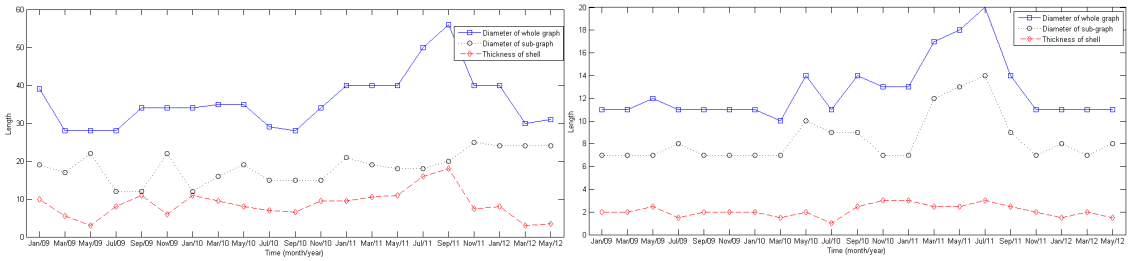
[2]They appear only as destination ASes, the last position in each AS routing path, and typically represent large corporations, universities, or Cloud/content providers

a very important reduction of the Internet routing table, as discussed in [58], T-E routing can lead to important improvements in terms of routing resiliency. Indeed, the introduction of many routing locators for the same destination drastically increases the Internet path diversity. If adequately managed for traffic engineering, the enlarged path diversity can lead to significant improvements of the Internet resiliency, as explained in [7] where a framework for coordinated edge-to-edge load-balancing and Internet-wide multipath routing is presented.

Therefore, new tools for Internet traffic engineering — currently limited to BGP tweaking practices such as prefix de-aggregation and transient announcements that are increasing the routing table size and are decreasing the Internet service reliability — could arise from T-E routing. At present, the potential achievable performance improvements are attracting attention from content providers and content delivery edge networks, especially with the emergence of Cloud Computing applications that require high connection resiliency and persistent reachability [7]. In the following, we characterize current interconnection, routing and traffic engineering practices of edge and transit ASes via measurement of BGP routing tables.

6.3 Interconnection Topology Analysis

The AS level Internet topology has been an active research area since at least one decade ago or so, and a major research efforts have been devoted to disclosing its characteristics [59]. In this paper, we choose to use BGP Routeviews' routing tables to proceed our analysis. BGP Routeviews' routing tables are captured from ASes that peer with many large transit carriers, so they represent a transit view on the Internet routes. Meanwhile, as far as we know, rare web servers or laboratories provide the AS interconnection information from the directional perspective of edge ASes. Therefore, it appears appropriate to represent routing maps using an undirected graph. Through studying the undirected graph, we first dissect the AS path properties, and then we observe edge and transit AS behaviors with several



(a) With AS path prepending (b) Without AS path prepending

Figure 6.1: The diameters of AS graphs vs time

graph theory metrics.

6.3.1 Path Properties

The AS-level Internet path can be considered as a key factor to quantify the interconnection behavior of AS networks. In practice, any AS can increase the AS path length artificially by repeating its own AS number, which is so-called AS path prepending [60]. To have a comprehensive view about the path properties, we approach our studies under two scenarios: without AS path prepending and with AS path prepending. We characterize the path properties in terms of the following aspects.

Diameter Diagnosis

The diameter of a graph is defined as the maximum shortest path length between any pair of nodes in the graph. If there is no path connecting two nodes, the diameter is set to infinity. It is a metric that reflects the connecting efficiency of the graph. In our studies, we quantify connecting efficiency of the Internet by diameter metric; Fig. 6.1a and Fig. 6.1b are the results with and without considering the AS path prepending, respectively (with path prepending, an AS artificially increases its AS paths by padding them with repetitions of its own AS number). The whole graph represents the whole Internet, the sub-graph is the transit networks and the thickness of shell is the half of the difference between the diameter of the whole graph and that of the sub-graph, which can represent the interconnection

status between edge and transit ASes.

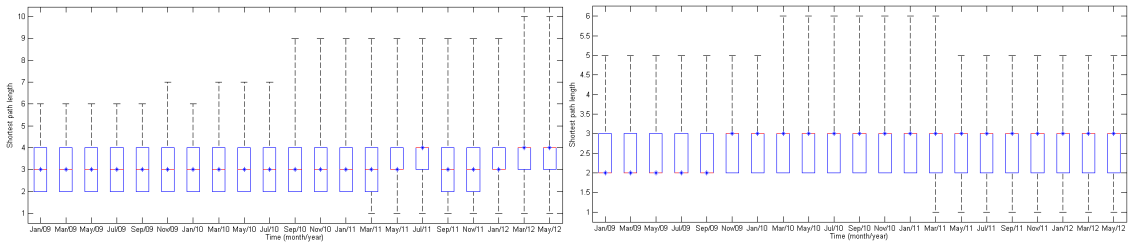
Obviously, the diameters are finite since the Internet is a whole interconnected network, hence its representation graphs are connected. With AS path prepending, the variances of the two diameters as well as the thickness of the shell become larger, and the diameter of the whole graph achieved the peak value around July 2011. Under the scenario of not considering AS path prepending, the diameter of the whole graph increased slowly before July 2011 and dropped dramatically since then, while that of the sub-graph shared a similar behavior from 7 to 14 and back to 7. The thickness only changed in the range from 1 to 3 during this time period.

The statistic results show us that:

- Without AS path prepending, the interconnection status among transit ASes has been improved since July 2011, which managed to control the increase trend of the two diameters.
- Without AS path prepending, the thickness of the edge shell did not change significantly, and the interconnection status between edge and transit ASes maintain the same level in the time period.
- With AS path prepending, the two diameters as well as the thickness of the shell reflected a certain degree of randomness.
- AS path prepending altered the AS graphs significantly.

Shortest paths diagnosis

We use the shortest paths between two edge ASes to analyze the potential inter-AS level routing efficiency from the perspective of edge networks. We choose 10% of the edge ASes that consistently "act "as edge ASes since January 2009. Then we measure and monitor the shortest paths between each pairs of the chosen ASes in the three years and a half period. We use boxplots to depict the results (each box, between the minimum and the



(a) With AS path prepending

(b) Without AS path prepending

Figure 6.2: Edge pairs shortest paths vs time

maximum, displays the first quartile, the median with a '*', third quartile). As we hope to reduce the importance of the outliners in AS shortest paths measurement, we treat the medians as the expected values of the shortest paths. Fig. 6.2a and Fig. 6.2b are the results with and without considering the AS path prepending, respectively. Though the maxima of the shortest paths change from time to time, the medians, the first and the third quartiles remain constant within each figure. When comparing the two figures, we find that the medians and the first quartiles within the two figures are almost the same all the times, while quartiles only increase 1 with AS path prepending. From the observations we can infer that:

- The potential performance of inter-AS level routing remain at the same level from the viewpoint of edge networks, although the Internet grows perpetually.
- For some edge network, the usage of AS path prepending enlarges the distance between them and some other edge networks.
- For most edge networks, AS path prepending actually does not degrade the potential efficiency of their global routing as long as a proper routing scheme can be designed and deployed.

6.3.2 Edge and Transit ASes Interconnection Properties Comparison

From the perspective of interconnection topology, edge and transit ASes hold dramatically different properties in the undirected graph. Next, we characterize the properties of the two

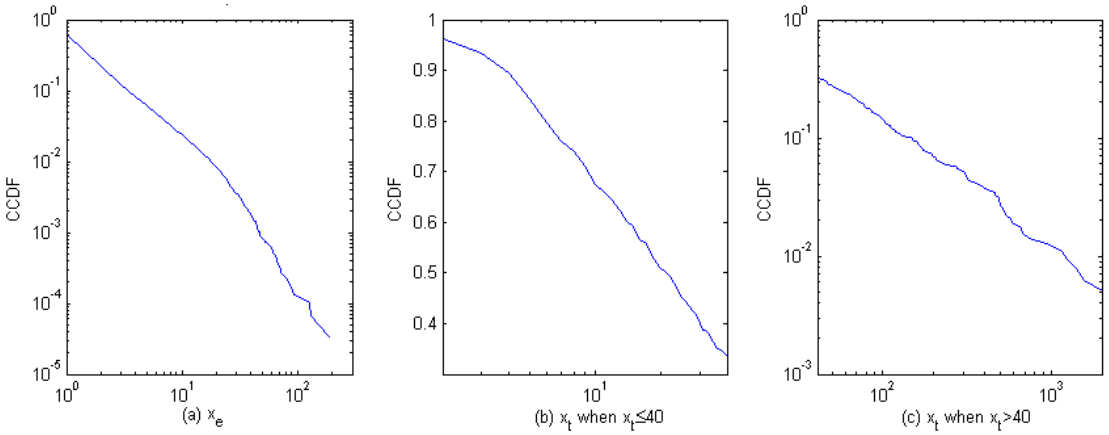


Figure 6.3: The degree CCDF of edge and transit ASes

types of ASes in the following aspects.

Degree analysis

The AS degree, defined as the number of AS neighbors, somehow reflects the importance of an AS. In Fig. 6.3 we plot the complementary cumulative distribution function (CCDF) of the AS degree for edge and transit ASes.

Let x_e and x_t denote the degree of edge and transit ASes, respectively. The CCDFs in Fig. 6.3 are obtained by analyzing the routing tables of Janary 2009, knowing that the same profile is approximately maintained for successive routing tables. Note that Fig. 6.3(a) and Fig. 6.3(c) use a log-log scale, while Fig. 6.3(b) uses a log-linear scale. We can see that the x_e CCDF linearly decreases in a log-log scale, and so does the x_t CCDF when the degree is bigger than a relative large threshold, e.g., 40. When x_t is smaller than the threshold, the CCDF decreases almost linearly in a log-linear scale. It is worth recalling that the CCDF of a nonnegative random variable that follows truncated discrete power law distribution[3] can be calculated as $F_c(x) \sim ax^{-\alpha}$, while the CCDF of a random variable that has truncated probability density function (pdf) as $f(x) = b/x$ can be calculated as $F_c(x) \sim -b\ln(x)$.

[3]Power law distribution has been observed in many fields for some time, especially in a wide variety of natural and man-made phenomena, and some physicists believe it corresponds to certain "universal laws" [61].

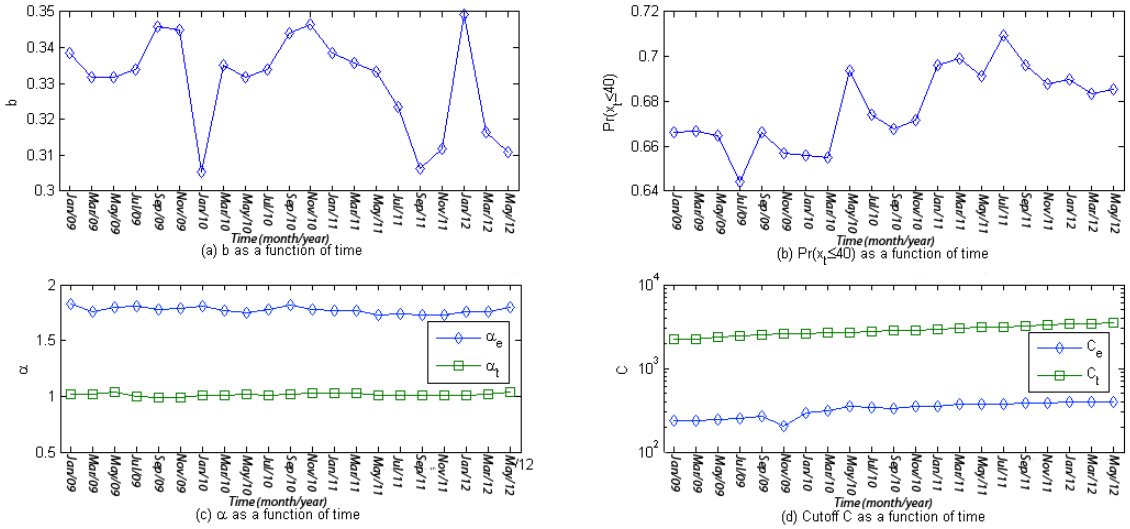


Figure 6.4: Model parameters vs time

In the following, we define the distribution with PDF $f(x) = b/x$ as inverse distribution; note that the CCDF of power law distribution becomes to linear function in a log-log scale, while that of inverse distribution shows linear characteristic in a log-linear scale. When combining the above results, we find that:

- The degree of edge ASes can be well fit with a power law distribution.
- When the degree of a transit AS is relatively small, it follows a truncated inverse distribution.
- When the degree of a transit AS is larger than a certain threshold, it follows a power law distribution.

To simplify the following analysis, we treat x_e and x_t as continuous random variables. Let the CCDFs for the degree of edge and transit ASes be F_{ce} and F_{ct}, respectively. We investigate the following relations:

$$F_{ce}(x_e) \sim a_e {x_e}^{-\alpha_e} \tag{6.1}$$

$$F_{ct}(x_t | 2 \leq x_t \leq d) \sim -b \ln(x_t) \tag{6.2}$$

$$F_{ct}(x_t|x_t > d) \sim a_t {x_t}^{-\alpha_t} \tag{6.3}$$

Please note that in (6.1) and (6.3) the CCDFs have right hand side cutoffs C_e and C_t, respectively.

From (6.2), we find $f_t(x_t|2 \leq x_t \leq d) \sim b/x$. As $\int_2^d f_t(x_t|2 \leq x_t \leq d)dx = 1$, we get:

$$b \sim \ln^{-1}(\frac{d}{2}) \tag{6.4}$$

Hence, as long as d is a constant, b as well as the statistics of x_t given $2 \leq x_t \leq d$ will also be deterministic. Through a similar derivation, the relationship between a and α can also be found.

In order to inspect the parameter trends, we choose $d = 40$, and apply the least square error (LSE) as the model estimator to the three years and a half time period routing tables. We first examine the trend of b to validate our previous analysis. From (6.4), we know that b should be around 0.33 given $d = 40$. The theoretical analysis perfectly fits our measurements reported in Fig. 6.4(a).

Next, we are interested in the trends of $Pr(x_t \leq 40)$, α_e, α_t, as well as the cutoffs C_e and C_t. In Fig. 6.4(b), we find that $Pr(x_t \leq 40)$ is relatively stable, which represents the probability for the degree of an transit ASes to follow power law distribution or inverse distribution is relatively stable. Fig. 6.4(c) shows that α_e is larger than 1.5 and smaller than 2, while α_t is very close to 1. Fig. 6.4(d) shows that C_t is much larger than C_e, and C_t as well as C_e have clear increasing trend in the time period since January 2009. Before further analyzing the results, let us discuss the properties of truncated power law distribution with pdf $f(x) \sim r x^{-\alpha-1}$ and two cutoffs c_1 and c_2 (c_1 is the left hand side cutoff, and c_2 is the right hand side cutoff). We only consider the case that $c_2 >> c_1$ and c_1 is 1 or 2. It is easy to show that:

$$E(x) \sim r \frac{c_2^{1-\alpha} - c_1^{1-\alpha}}{1-\alpha} \tag{6.5}$$

$$E(x^2) \sim r \frac{c_2^{2-\alpha} - c_1^{2-\alpha}}{2-\alpha} \tag{6.6}$$

When $\alpha \approx 1$, based on (6.5), we can get the equation

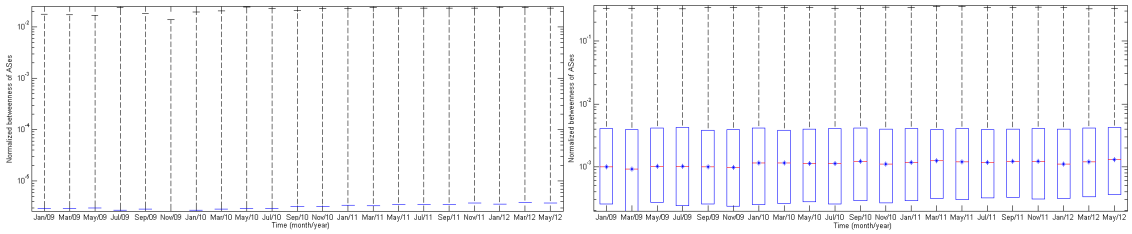
$$\lim_{\alpha \to 1} E(x) \sim r \ln(c_2) \tag{6.7}$$

Combining the observations, we can assert that:

- The average of x_t is increasing in these years, as α_t is very closer to 1 and the cutoff C_t is always raising. This shows the interconnection of transit ASes evolves permanently, by which a lot of new shortest paths can be created to improve the performance of the Internet.
- Following the raise of cutoff C_e, the average of x_e is also increasing in these years. This reflects the fact that edge networks are increasingly performing upstream multi-homing to improve the network interconnection situation.
- Based on (6.5)-(6.7) and simple calculations, we can find that the standard deviations of x_e and x_t are also increasing in last years. This indicates that the distributions for the degree of edge and transit ASes are stretching constantly.

Betweenness Diagnosis

The centrality of a node within a graph can be measured by its betweenness [62], which is calculated by counting the number of all the possible shortest paths passing the corresponding node. In practice, one usually normalizes the betweenness with the total number of the shortest paths to get so-called normalized betweenness. The normalized betweenness



(a) Edge ASes (b) Transit ASes

Figure 6.5: The normalized ASes betweenness vs time

of an absolute center, through which almost every shortest path would go, should equal or very close to 1. In order to minimize the impact of the Internet growth to our analysis, we apply normalized betweenness to gauge the centrality of each AS. We still utilize boxplots to depict the data and use the star symbol to emphasize the medians of the data for each box.

Fig. 6.5a and Fig. 6.5b are the boxplots for the betweenness of edge and transit ASes, respectively. In Fig. 6.5a, only the third quartile and the maximum value can be seen, as other statistics are too small to be shown. Fig. 6.5a shows that at least 75% of edge ASes have extremely small betweenness. In Fig. 6.5b, the first quartiles are around $2 * 10^{-4}$, while the maxima changes around 0.3. These statistics show that

- Compared with edge ASes, transit ASes usually hold a much bigger degree of centrality.
- Most transit ASes do not have high centrality, and they do not play the role of central ASes in the Internet presently.
- Certain transit ASes do have a very large betweenness, and is constantly around 0.3.
- From the viewpoint of graph theory, some transit ASes are of very importance and serve as partial centers to the Internet, and the misbehavior of these ASes may affect 30% of the inter-AS routing decisions.

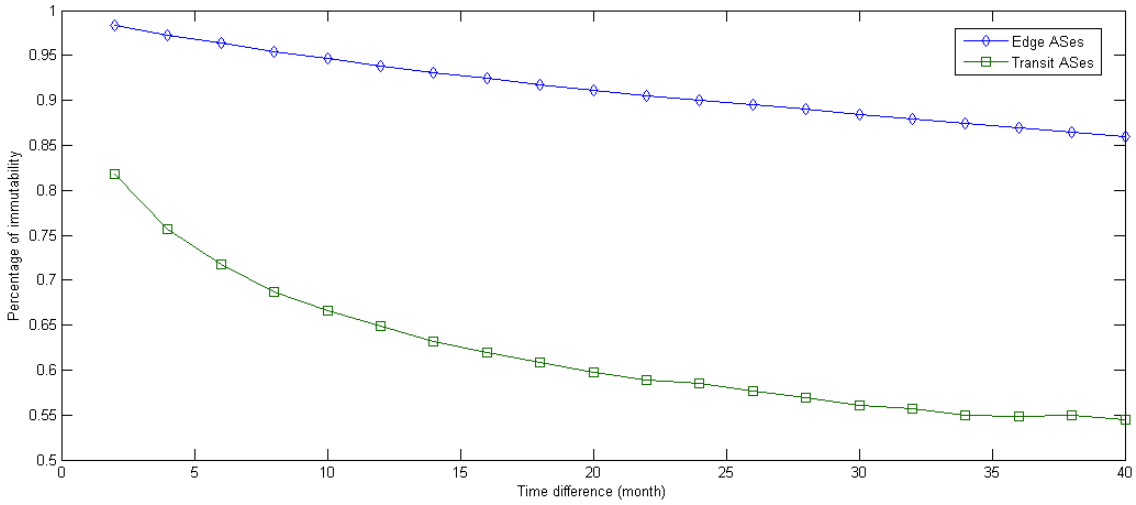
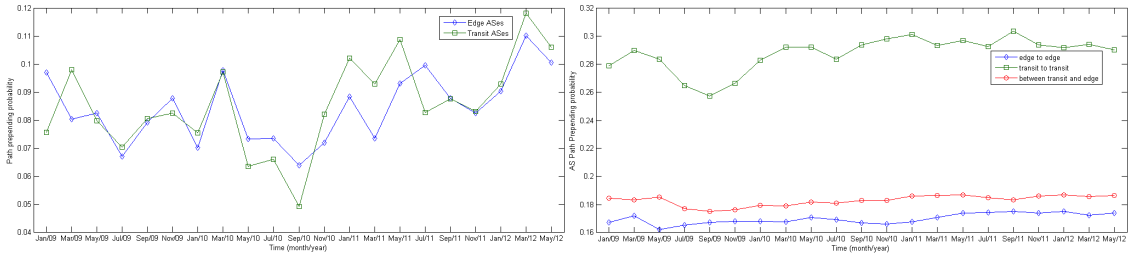


Figure 6.6: The roles immutability of ASes vs time difference

T-E Routing Properties

According to the position of each AS in the routing entries, the Internet can be artificially categorized into edge and transit networks; obviously, an AS should not hold both roles at the same time. However, the role of a particular AS may change abruptly, due to interconnection evolution or routing fluctuations; this phenomenon is shown in Fig. 6.6 (filtering out path prepending). The horizontal axis represents the time difference, and the vertical axis represents the percentage of a kind of ASes that still hold their original ranking after the time interval (defined as AS role immutability). From Fig. 6.6, we can see the immutability of edge networks decreases almost linearly from 98% to 85% when the time difference increase from 2 months to 40 months, while the immutability of transit networks decreases in a convex way from 81% to 55% which seems approaching certain value as the time difference becomes larger. Given these observations, we can state that:

- The roles of ASes are quite immutable in a short relative period, like 1 or 2 months.
- The immutability of edge ASes is higher than that of transit ASes.
- The immutability of edge ASes decreases linearly while that of transit ASes decreases in a convex that as the time difference increases, and seems certain transit ASes would



(a) AS node (b) AS link

Figure 6.7: AS path prepending probabilities vs time

never change their roles.

- T-E routing categorization should not rely on an automated detection of current roles, but should be set statically by transit ASes with little or no coordination with edge ASes.

Such role changes indicate that edge ASes rarely "evolve" as transit ones, but rather the inverse occurs, i.e., ASes in the transit core are pushed towards the edges as the time passes.

6.4 Routing and Traffic Engineering Analysis

In this section, we characterize edge and transit networks from a routing and traffic engineering point of view. Among all the available traffic engineering techniques in BGP routing, we can mention local preferences for outbound traffic engineering, AS path prepending for inbound traffic engineering, and IP de-aggregation for multi-homing traffic engineering. While the first cannot be inferred with adequate precision from routing table analysis, path prepending and IP de-aggregation can, as reported in the following. Such practices coupled with the BGP convergence issue indirectly affects the BGP routing instability, which is an aspect also analyzed in this section.

6.4.1 AS path prepending analysis

With AS path prepending, artificially repeating its own AS number to increase the length of certain AS paths towards itself, an AS can meet inbound traffic engineering goals, i.e., distracting incoming traffic toward more available or preferred entry points. We are interested in the occurrence of path prepending, including the empirical probability for an AS to apply path prepending, as well as for an AS link to be affected by path prepending. We categorize the AS links into three types: links within edge networks, links between edge and transit networks and links within transit networks. Fig. 6.7a shows the experimental probabilities that edge and transit ASes use path prepending, while Fig. 6.7b shows the probabilities that the three types of AS links are affected by path prepending. In Fig. 6.7a, we find that not only are the probabilities to employ AS path prepending very close to each other, but they also share the same time profile. In Fig. 6.7b, we see that the AS links within transit networks are affected by path prepending with the highest probability while the links within edge networks are with the lowest probability. All in all, we can assert that:

- The path prepending occurrence for edge and transit ASes is relatively low, as it is below 0.12 for both.
- The occurrence probabilities for edge and transit ASes to apply AS path prepending are very similar with each other.
- The transit networks perform inbound traffic engineering more frequently than edge ASes.

Edge ASes apply path prepending essentially for inbound load balancing, while transit ASes perform path prepending as a second-level routing rule for provider transit vs. client transit and transit links vs. peering links load-balancing (the first-level rule for such operations typically is the local-preference).

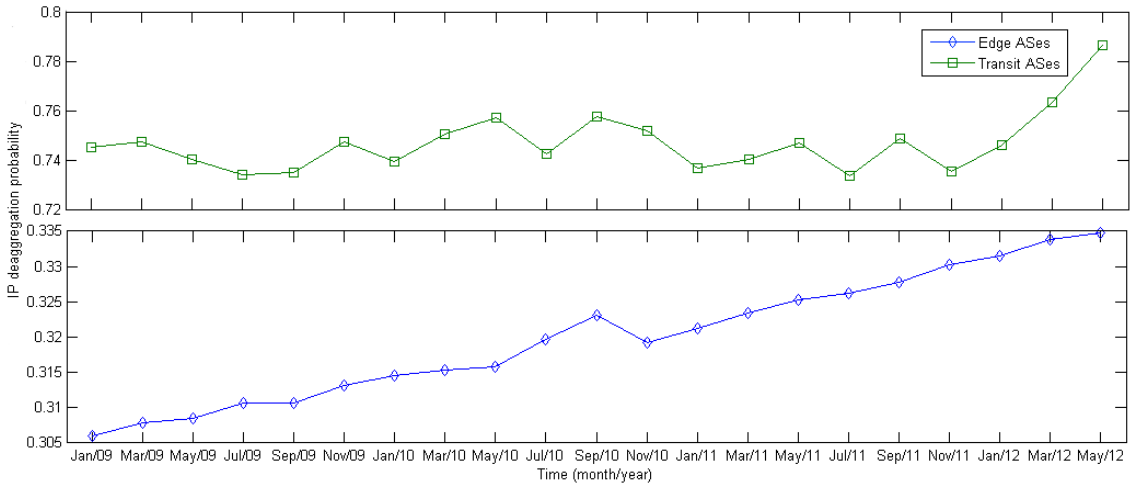


Figure 6.8: ASes IP de-aggregation probabilities vs time

6.4.2 IP de-aggregation probability diagnosis

For security, resiliency as well as load balancing purposes, ASes can artificially fragment large IP prefixes into several smaller prefixes and announce them separately [63], [64]. This behavior is usually known as IP prefix de-aggregation. Although both transit and edge networks may employ this technique to meet certain goals, due to the difference between their functions in inter-AS routing, the probabilities of their usage as well as the specific usage behaviors may be different. In our analysis, we gather all the IP prefixes that announced by the same AS, and use seamless and precise IP aggregating rule to check if the AS utilize IP de-aggregation or not. For instance, suppose an AS announce 1.2.3.128/25 and 1.2.3.0/25 separately. As 1.2.3.128/25 and 1.2.3.0/25 can be aggregated into 1.2.3.0/24, we deem that the AS applies IP de-aggregation.

Fig. 6.8 shows the probabilities for edge and transit ASes to apply IP de-aggregation. We find that the de-aggregation probability of edge ASes has a clear increase trend, while that of transit ASes oscillate between 0.73 and 0.79. These properties tell us that:

- IP de-aggregation is very a popular technique among edge and transit ASes in these years.

- More and more edge ASes are increasingly applying IP de-aggregation currently, and that imposes more pressure to the scalability and efficiency of the global routing.
- Compared with edge ASes, transit ASes are more active in utilizing IP de-aggregation to meet traffic engineering requirements.

6.4.3 Prefix de-aggregation impairment analysis

We analyze the impairment of IP prefix de-aggregation at time t in BGP routing tables in the following way: first, we gather all the IP prefixes announced by a given AS x at time t, noting the total number of prefixes as d_{xt}; next, we recursively apply a seamless and precise IP aggregating rule to obtain the size of the IP prefixes before IP de-aggregation, which is noted as a_{xt}; then the IP de-aggregation rate r_{xt} of the AS x can be expressed as:

$$r_{xt} = \frac{d_{xt} - a_{xt}}{a_{xt}} \tag{6.8}$$

For example, an AS announces 1.2.3.128/25, 1.2.3.0/25 and 128.1.1.0/24, separately; as 1.2.3.128/25 and 1.2.3.0/25 can be aggregated with 1.2.3.0/24, the de-aggregation rate of the AS is $(3-2)/2 = 0.5$. Therefore, any AS that does not employ IP de-aggregation should have a zero IP de-aggregation rate.

Fixing the total number of ASes to N, an AS that can communicate with every announced prefix at time t should have a BGP routing table size close to $\sum_{i=1}^{N}(a_{it}r_{it} + a_{it}) = \sum_{i=1}^{N} a_{it}r_{it} + \sum_{i=1}^{N} a_{it}$. Nevertheless, in an ideal scenario, if there is no IP prefix de-aggregation, its BGP routing table size should only be $\sum_{i=1}^{N} a_{it}$. Due to IP prefix de-aggregation, the routing table size gets indeed significantly enlarged.

If we consider the overall impact of IP de-aggregation to the sizes of routing tables and let

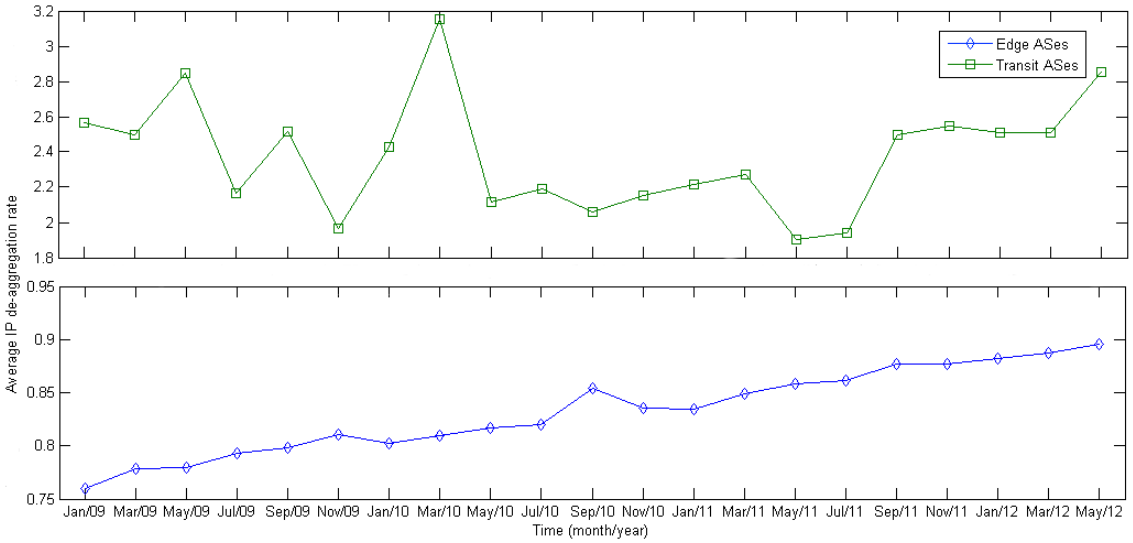


Figure 6.9: The average of ASes prefix de-aggregation rates vs time

R_t be the impact ratio of the routing tables, then:

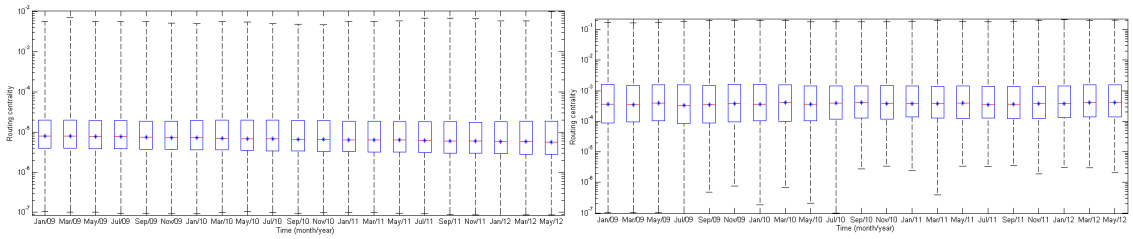
$$R_t = \frac{\sum_{i=1}^{N} a_{it} r_{it}}{\sum_{i=1}^{N} a_{it}} \tag{6.9}$$

where, $i \in [1, N]$, a_{it} are unknown constants and r_{it} can be treated as independent identically distributed (IID) random variables due to the partial arbitrary nature of IP prefix de-aggregation. From (6.9), we know that:

$$E(R_t) = E(r_{it}) \tag{6.10}$$

Therefore, if we could find an alternative routing mode with some form of hierarchical routing more natively supporting IP prefix de-aggregation – such as a T-E routing protocol – while allowing at least the same level of traffic engineering capabilities, the BGP routing table size could shrink dramatically.

The average prefix de-aggregation rates for edge and transit ASes are shown in Fig. 6.9. We find that for edge ASes it increases quite clearly in time, while for transit ASes it oscillates



(a) Edge ASes (b) Transit ASes

Figure 6.10: The normalized ASes routing centrality vs time

in time. The overall IP prefix de-aggregation rate, mainly decided by edge ASes because of their much larger total number, has grown from 0.79 to 0.94 in the three years and a half period, which further stresses the Internet scalability (higher impact on routing tables). From the studies, we can assert that:

- Transit ASes are more used to IP prefix de-aggregation than edge ASes, and its de-aggregation usage can vary significantly in time and not necessarily increases, while edge ASes usage de-aggregation raises constantly.
- For edge ASes, there is an increasing use of IP prefix de-aggregation across the years.
- The IP de-aggregation rates of edge and transit ASes directly impair the scalability and efficiency of the Internet, and the average value of impact ratio R is affected by the de-aggregation rate r_{it} (6.10).
- Following the growth of the overall prefix de-aggregation rate, the impairment of prefix de-aggregation also increases in these years.

All in all, it is worth stressing that one would expect that edge networks do not perform actively traffic engineering because of the much lower scale of bitrate aggregates than for transit networks. However, we have verified that not only they actively do incoming and multi-homing traffic engineering (via BGP path prepending and prefix de-aggregation), but that they do that at a close level to the level at which transit networks do.

6.4.4 Routing Centrality Comparison

Sparked by the definition of betweenness in graph theory, we use the appearance time of an AS in the routing table to measure the routing centrality of the AS. For each AS, we count the number of times the AS appears in the routing table, and normalize the final count by the table size to get the normalized routing centrality. So the normalized routing centrality of an absolute routing central AS, which almost appears in every routing entry, should equal or very close to 1. We use boxplots to depict the normalized routing centrality statistics of edge and transit ASes in Fig. 6.10a and Fig. 6.10b, respectively. In Fig. 6.10a, the third quartiles change around $2*10^{-5}$, while the medians are $7*10^{-6}$. In Fig. 6.10b, the first quartiles changes around 10^{-4}, the medians are $4*10^{-4}$, and the maxima are round 0.2. All in all, we can assert that:

- The expected normalized routing centrality of a transit AS is almost 50 times larger than that of an edge AS.
- The normalized routing centralities of all the edge and transit ASes are far below 1, which reflects that there is no absolute routing central AS nowadays.
- The normalized routing centralities of some transit ASes are constantly around 0.2. Hence, some ASes hold a particular large normalized routing centrality in the Internet currently.

Here, we achieve a similar conclusion with what we got from the analysis of betweenness, which tells us that some transit ASes are very vital in the global routing. The failure of those ASes may result in a series of severe impairments, e.g., consuming enormous routers resource to converge new routing tables, evoking huge time delay, degrading the routing efficiency of the Internet, etc.

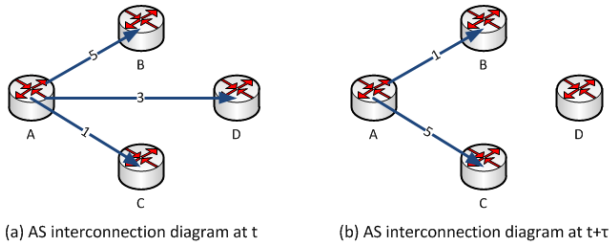


Figure 6.11: AS interconnection diagrams

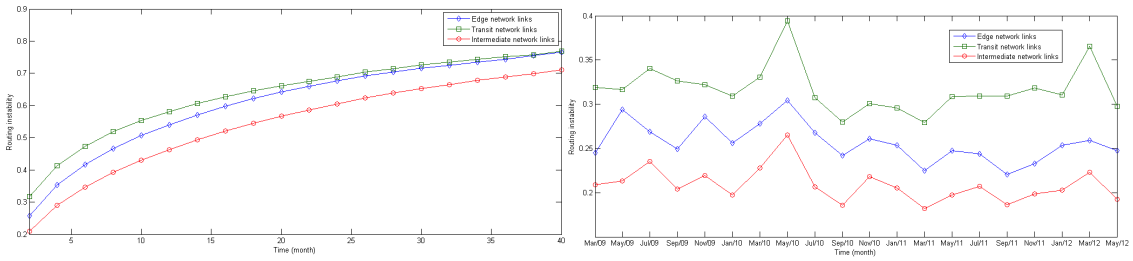
6.4.5 Routing Instability Analysis

Internet routing instability represents the fluctuation of routing information towards network reachability. Many reasons are behind this phenomenon, including the change of infrastructure, the impact of traffic engineering, the employment of multi-homing, etc. However, high levels of routing instability can lead to serious impairments, e.g., packet loss, increase of network latency and time to convergence, and even the loss of interconnection availability in wide-area or national networks [65].

In inter-domain routing, routing instabilities can be roughly characterized from the fluctuation of the BGP routing table. In the following, we define the appearance time of an AS-level link i in a routing table as the occurrence count of the link, also define the average of the overall change rate as the routing instability rate, noted as RI. We consider RI as an adequate metric to quantify the routing instability. If we represent an undirected graph at time t with $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, where $\mathcal{V}_t$ is the set of the nodes and $\mathcal{E}_t$ is the set of links, the RI after time τ can be calculated as follows:

$$RI = \frac{1}{|\mathcal{E}_t|} \sum_{i \in \mathcal{E}_t} \frac{|n_i^t - n_i^{t+\tau}|}{\max(n_i^t, n_i^{t+\tau})} \tag{6.11}$$

where, $|\mathcal{E}_t|$ is the size of the link set, n_i^t is the occurrence count of link i in the routing table at time t, and $n_i^{t+\tau}$ is the occurrence count of link i in the routing table at time $t + \tau$. If link i cannot be found in the routing table at time $t + \tau$, we set $n_i^{t+\tau} = 0$.



(a) Time difference (b) Time

Figure 6.12: Routing instability vs time

A demonstration of how to use (6.11) is shown here. Suppose we want to calculate the RI between Fig. 6.11(a) and Fig. 6.11(b), then $RI = 1/3*(|5-1|/5+(3-0)/3+|1-5|/5) \simeq 0.87$. As there is considerable difference between Fig. 6.11(a) and Fig. 6.11(b), we get a very big RI, which represents the routing instability between the two graphs is in a significantly high degree.

We now categorize AS links into three types: edge network links, which connect edge ASes, transit network links, which connect transit ASes, as well as intermediate network links, which connect edge and transit ASes. Then we use (6.11) to measure the routing instability status of these three types of links, which are shown in Fig. 6.12. In Fig. 6.12a, the horizontal axis is the time difference τ and the vertical axis is the routing instability given the time difference τ. In Fig. 6.12b, the horizontal axis is the time t, and the vertical axis is the routing instability between the routing table at time $t - \tau$ and the routing table at time t on a fixed time different $\tau = 2$ months. We find that the routing instabilities of the three types of links all raise gradually in a similar way when the time difference increases. When the time difference is fixed at two months, the routing instabilities of the three types of links also vary with a similar pattern.

From the two figures, we can assert that:

- The routing instabilities of the three types of links all raise in concave fashions as long as the time difference increases.
- Among the three types of links, the routing of intermediate network links is the most

stable, while that of transit network links is the least stable.

- If the time difference is large than certain value, i.e., 35 months, the routing instabilities of edge network links and transit network links will be extremely similar.
- When the time difference is fixed at two months, the routing instabilities of the three types of links also share the similar pattern as time changes.
- Currently, the routing instability phenomenon is relatively serious, as the minimum value in the two figures is still around 0.2.

Two main factors can be behind such a routing instability: the inner convergence and oscillation problems of BGP, and the practice of edge and transit networks in performing inbound and outbound traffic engineering operations.

6.5 Measurement Remark

Transit-edge routing functionally proposes to create a two-level hierarchical routing between networks that have different routing behavior. In this section, we measure real inter-domain routing information to characterize behavior and properties of edge and transit AS networks with a transit-edge hierarchical routing perspective.

From an interconnection point of view, we first analyze the diameters and the shortest paths between ASes pairs. We unravel that although the Internet grows constantly and AS path prepending impact the structure of the Internet significantly, the Internet service performance for most edge network would not degrade as long as a proper routing scheme can be deployed. Next, we found that the interconnection degree of an edge AS can be well fit with truncated power law distribution, while that of a transit AS can be fit by the combination of power law and inverse distribution, and we analytically and experimentally identified the different regimes of edge AS and transit AS degree distributions. From a routing and traffic engineering viewpoint, we discovered that edge and transit ASes have similar probabilities of applying AS path prepending. We categorized the AS links into three types, and

unraveled that they are affected by path prepending with different probabilities. We also discovered the facts that edge and transit ASes have similar probabilities of applying AS path prepending, while transit ASes are more prone to utilize IP de-aggregation. We recognized that the impact ratios of BGP routing tables are directly determined by the IP prefix de-aggregation rate of edge and transit ASes, discovering that transit ASes do de-aggregate their own prefixes 3-times more often than edge ASes, which may appear surprising and counter-intuitive. Moreover, we described a mechanism to measure the routing instability phenomenon, recognizing that the transit network links have the largest routing instability while the intermediate network links have the least routing instability[4].

From a traffic engineering requirement perspective, one would expect that edge networks do not actively perform traffic engineering because of the much lower scale of bitrate aggregates than for transit networks. However, we have verified that not only edge ASes actively do traffic engineering, but that they do it at a close level to the one at which transit networks do. Moreover, it appears that the multi-homing traffic engineering trend, based on prefix de-aggregation, is a practice increasingly adopted by edge ASes.

[4]Implementation and codes are given in [43].

Chapter 7: Conclusion

In this dissertation, a generalized multipath routing framework is presented to solve the multipath routing problem.

Through studying and categorizing the costs for source and destination nodes, the interaction between the two distant independent nodes can be modeled as a non-cooperative game. When only considering single metric, the game can be further expressed as a cardinal potential game. Not only does this characteristic simplify the process of finding the NEs but also brings a preliminary multipath routing framework by using the potential value to evaluate the routing strategies.

In addition, a novel vectorized routing cost model, based on a vector space and game theory, is defined to overcome the limitation of the previous model. With the vectorized routing model, the framework is capable of considering multiple metrics at the same time.

To solve the vectorized routing model, a set of universal refinement tools is proposed, and one of them is proved as the extensive form of the potential value method. Through the refinement tools, a generalized multipath load sharing framework is achieved. The generalized routing framework is applicable to more general settings since it does not count on specific characteristics of the game, which brings an even wider practice scope for the load sharing framework.

Through multi-criteria simulations, it is obvious that the generalized load sharing framework is able to offer far more resilient solutions with respect to multipath BGP as well as the basic routing model of the LISP protocol currently under standardization.

Meanwhile, we measured the Internet from inter-domain routing viewpoint to comprehend the overall routing behaviors inside the Internet for inter-domain routing. Similar to what

was predicted, a lot of efforts and diverse traffic engineering practices indeed have been utilized by different Autonomous Systems to achieve resiliency Internet service. Among those efforts, some actually would undermine the healthy and productive evolvement of the Internet, i.e., multi-homing and IP prefix de-aggregation, thus it is necessary, or even urgent, to find an alternative way to fulfill the resiliency requirement.

Based on the popularity of multi-homing behavior among edge domains, we find it is possible to increase the resiliency through exploiting the path diversity brought by multi-homing. In addition, when edge domains are theoretically treated as edge nodes, the generalized multipath framework, proposed in this dissertation can be applied.

The model has the potential to be generalized to multiple network-player settings. However, the introduction of multi-player may also bring in the issue of congestion, as multiple sources may choose the same locator for the same destination. Thus the model needs to be improved to also consider the cost of potential congestion. In addition, how to consider and model the interaction between upstream and downstream flows is also an interesting extension.

Meanwhile, the path evaluation method proposed in this dissertation mainly depends on path diversity as well as individual paths' length. More sophisticated methods could be developed to consider other factors, i.e., path capacity, path congestion, path stability, etc. The quality of the path may be modeled by some tricks, such as artificially lengthening or shrinking the correspond path length. When dynamically depicting the status, certain stochastic processes may be introduced, which would make the model more practical and attractive.

A new mechanism, called BGP Link State (BGP-LS), is proposed recently to facilitate the distribution of current state of the connections within the network on BGP layer using BGP itself instead of IGPs. It can be expected that BGP-LS would enhance the traffic engineering practices on BGP layer, including the proposed multipath load sharing framework. However, further study is still necessary, e.g., how to design the interaction between the framework and BGP, when to initial load sharing, what type of information should be collected, etc.

Appendix A: Prisoner Dilemma and Potential Games

We provide in this appendix a brief "tutorial" on how to decompose a prisoner's dilemma game as sum of two interesting types of games. Consider the generic symmetric game in Table A.1, where $a, b, c, d \in \mathbb{R}$. We have a prisoner dilemma *cost* game if $a > b > c > d$, with (B, R) as Nash equilibrium, inefficient since both would prefer (T, L), which is however a dominated strategy profile. Indeed, this is the rationality dilemma offered by such games.

Table A.1: A generic 2-player symmetric game

I\II	L	R
T	(c, c)	(a, d)
B	(d, a)	(b, b)

The game can be decomposed as sum of the two games shown in Table A.2. For the first game, the cost components for the two players are equal for every profile. For the second game, the cost components of a player do not depend on its choice, but they depend on the other player's choice. The second game can be called "dummy game" since for a player there is no possible discrimination in choosing one strategy instead of the other. It can also be called "game of pure externality" meaning that its action has an effect *only* on the other player. This type of decomposition allows to clearly see the externality effect in the prisoner dilemma game.

Table A.2: Decomposition of a 2-player symmetric game

I\II	L	R
T	$(0, 0)$	$(d-c, d-c)$
B	$(d-c, d-c)$	$(d-c+b-a, d-c+b-a)$
I\II	L	R
T	(c, c)	$(a-d+c, c)$
B	$(c, a-d+c)$	$(a-d+c, a-d+c)$

Table A.3: Decomposition of a 2-player symmetric game

I\II	L	R
T	$(0,0)$	$(-1,-1)$
B	$(-1,-1)$	$(-2,-2)$
I\II	L	R
T	$(2,2)$	$(5,2)$
B	$(2,5)$	$(5,5)$

With the setting: $a = 4, b = 3, c = 2, d = 1$ we obtain the game decomposition in Table A.3. The choice of B allows to decrease the cost of I by 1, independently of the choice of II. At the same time, this choice increases by 3 the cost of II in the second game. It is worth noting that, in the first game, the costs are equal for the two players and that the choice of B has a positive externality effect for II: the choice also decreases II's cost by 1.

With a broader perspective, one can note that such a decomposition is a general property of the so-called potential games [41]. For a game in strategic form $G = (X, Y; f, g)$, where X and Y are the strategy sets for the two players, and f and g are real functions, G admits a potential if it exists a function $P : X \times Y \to \mathbb{R}$ such that $\forall x', x'', x \in X, \forall y', y'', y \in Y$:

$$\begin{aligned} P(x', y) - P(x'', y) &= f(x', y) - f(x'', y) \\ P(x, y') - P(x, y'') &= g(x, y') - g(x, y'') \end{aligned} \tag{A.1}$$

P is called *potential function*. The analogy with physics relates, e.g., to the ability to substitute a "vector field" (the two payoff functions) with a single scalar valued function, or to the condition of being an irrotational field. Minima of the potential function are Nash equilibria for the game, which guarantees that finite potential games have equilibria in *pure* strategies [41].

Potential games emerge from congestion problems [66]. Indeed, we can represent the game of Table A.1 with Figure A.1. Both players have to go from *start* to *arrival* taking either path A or path B (strategy A corresponds to T for I and to L for II, B corresponds to B for

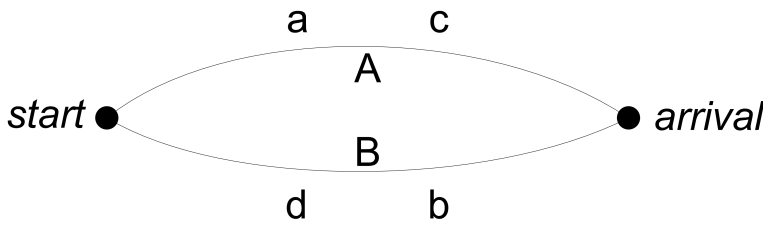


Figure A.1: Representation of a 2-player symmetric game

I and to R for II). The lower-case letters on each path in Figure A.1 indicate the forward cost for the players in case they walk alone (on the left) or together (on the right). If they travel together on the same path, the path is more congested than if they traveled alone along different paths, i.e., the cost is higher for both.

Appendix B: On Mixed Strategy Equilibria

In a non-cooperative routing game, a strategically acceptable way to seek an *arbitrary load-balancing distribution* (e.g., 24%, 47% and 29% for three locators) might theoretically be reached implementing "mixed strategy"equilibria that could appear in addition to pure-strategy equilibria (the "type"discussed so far).

It is worth doing a small digression on this aspect. In game theory, with mixed strategies the player no longer chooses a single strategy, but a probability distribution on its (unilateral) available strategies. Somehow the player can rely on a random process that implements his decision following the probability distribution. In non-cooperative games, players adopt independent random processes, and the probability distribution of a strategy profile (e.g., an equilibrium) is given by discrete multiplication of the probabilities each player assigned to its corresponding strategy. Note that an equilibrium in pure strategies can be seen as a particular (degenerated) equilibrium in mixed strategies where each player strategy, hence the strategy profile, has a probability equal to 1. For example, in the game of Table 3.3, the equilibrium strategy G_2L_3 is played by node A with probability $p = 1$ and the other five strategies with probability $1 - p = 0$, and the same for node B and the equilibrium strategy G_3L_2 played with probability $q = 1$, so that the equilibrium profile (G_2L_3, G_3L_2) is played with probability $p \cdot q = 1$.

In game theory parlance, this is quite straightforward once noted that the Nash equilibrium(a) of G can be found by iterated reduction of strongly dominated strategies. For example, in Table 3.3 the equilibrium can be obtained by first excluding, for node A, all G_1 strategies and G_2L_4 and G_2L_5 strategies since whatever node B chooses the A cost is always minor, and by then conversely excluding G_4 and G_5 and G_3L_1 strategies for B. The reduced game is the game degenerated to the single Nash equilibrium, if it is unique, and thus no mixed strategy is conceivable. If multiple equilibria exist for the general setting, the reduced game is composed of as much strategies and strategy profiles as needed to

encompass the equilibria, and no additional mixed-strategy equilibria arise.

Appendix C: Pareto Efficiency

It is worth recalling that the Nash equilibrium can be inefficient and far from the social optimum: the paid price is the price of anarchy due to the non-cooperative modeling of edge networks' independency. A strategy profile p is *Pareto-superior* to another profile p' if a player's cost can be decreased from p to p' without increasing the other players' costs. The *Pareto-frontier* contains the *Pareto-efficient* profiles, i.e., those not Pareto-inferior to any other. In our routing game, locator costs affect the Pareto-efficiency (because of the pure externality of G_d); In particular, given many Nash equilibria, their Pareto-superiority strictly depends on G_d. For example, in Table 3.3, the strategy profiles in italic are Pareto-superior to the Nash equilibrium, but are not equilibria since at least one player is interested in deviating to reduce its cost. Moreover, those underlined are the Pareto-efficient profiles of the game, and also correspond to the social optimum (which is not true in general). Hence the game has the form of a Prisoner-Dilemma game, where the players see the convenience to adopt a Nash equilibrium solution despite other non-equilibrium profiles are more efficient for both of them. Moreover, it is a good exercise to check that, if we decrease $c_{1,4}$ to 10, we obtain a second equilibrium in (G_1L_4,G_3L_2) which is Pareto-superior to the other equilibrium (G_2L_3, G_3L_2). This is due to the external effect of G_d, i.e., $c(l'_3) > c(l'_4)$.

Appendix D: Lemke-Howson Algorithm

D.1 Introduction

The Lemke-Howson algorithm is an effective method to find at least one Nash Equilibrium (NE) for a two-person bimatrix game. Here the conception of NE is extent to include not only pure NE but also mixed NE. The algorithm was first introduced in [46], and was interpreted geometrically in [48], which visualize the process of finding NE when both players' strategy sizes are small enough. The explanation here is based on the work in [48], and we also add certain our own understandings to make the statement even clearer.

D.2 The Lemke-Howson Algorithm

Consider a two-player game with payoff matrix as $U_i,\ i \in \{1, 2\}$ for player 1 and 2, respectively. We assume that all the elements in $U_i,\ \forall i \in \{1, 2\}$ are positive. The assumption is without loss of generality, since adding the same large number for every elements in $U_i,\ i \in \{1, 2\}$ clearly does not change the characteristic of the game, and one NE of the original game will still be a NE of the adjusted game.

Suppose player 1 has m strategies available, denoted as $S_1 = \{s_1, ..., s_m\}$, while play 2 has n strategies available, denoted as $S_2 = \{s_{m+1}, ..., s_{m+n}\}$. Therefore, U_1 and U_2 are both $m \times n$ matrix. Call a vector is a probability vector if it represents a probability distribution, i.e., all the elements are non-negative and the sum is 1.

Use p_i to represent the probability for player 1 to choose strategy s_i for $i \in \{1, ..., m\}$, while use q_j to represent the probability for player 2 to choose strategy s_j for $j \in \{m+1, ..., m+n\}$. For a pair of probability vector (p, q), $p = \{p_1, ..., p_m\}^T$ and $q = \{q_{m+1}, ..., q_{m+n}\}^T$, it is a Nash Equilibrium (NE) if and only if for player 1 either $p_i = 0$, or $p_i > 0$ and s_i is the best reply to q, while for player 2 either $q_j = 0$, or $q_j > 0$ and s_j is the best reply to p. Suppose

(p, q) is a NE, then:

$$\begin{aligned} U_1 \cdot q + r &= v_1 \cdot \mathbb{1} \\ U_2^T \cdot p + t &= v_2 \cdot \mathbb{1} \end{aligned} \tag{D.1}$$

where $r = \{r_1, ..., r_m\}^T$, $t = \{t_{m+1}, ..., t_{m+n}\}^T$ and $r \geqslant 0$, $t \geqslant 0$, while $\mathbb{1}$ indicates a column vector of 1's of appropriate dimension and $v_i,\ i \in \{1, 2\}$ is a scalar representing player i's payoff.

In (D.1), r satisfy the condition that $r_i \neq 0$ when and only when $p_i = 0$ for all $i \in \{1, .., m\}$ and t satisfy the condition that $t_i \neq 0$ when and only when $q_i = 0$ for all $i \in \{m+1, .., m+n\}$.

From (D.1), it is easy to get:

$$\begin{aligned} U_1 \cdot q' + r' &= \mathbb{1} \\ U_2^T \cdot p' + t' &= \mathbb{1} \end{aligned} \tag{D.2}$$

where $q' = q/v_1$, $r' = r/v_1$, $p' = p/v_2$ and $t' = t/v_2$.

Define the calculation of normalization for vectors as follows:

$$normal(x) = x / \sum_i x_i$$

Then, $normal(p') = p' / \sum_i p'_i = v_1 \cdot p' = p$. The same operation also stands for q', which

brings that:

$$\begin{aligned} p &= normal(p') \\ q &= normal(q') \end{aligned} \tag{D.3}$$

Therefore, it is easy to get p and q as long as we can find p' and q'.

For (D.2), one obvious solution is $p' = 0$, $q' = 0$, $r' = \mathbb{1}$ and $t' = \mathbb{1}$. This solution, called *extraneous solution*, is a by-product of (D.2) and apparently does not fit our original problem. However, the extraneous solution can be used to find a practical solution, and the process will be explained in detail later.

To find p' and q', let us rewrite (D.2) as follows:

$$[U_2^T \ I] \begin{pmatrix} p' \\ t' \end{pmatrix} = \mathbb{1} \tag{D.4}$$

$$[I \ U_1] \begin{pmatrix} r' \\ q' \end{pmatrix} = \mathbb{1} \tag{D.5}$$

Again, in (D.4) and (D.5), r' satisfy the condition that $r'_i \neq 0$ when and only when $p'_i = 0$ for all $i \in \{1, .., m\}$ and s' satisfy the condition that $s'_i \neq 0$ when and only when $q'_i = 0$ for

all $i \in \{m+1, .., m+n\}$, and these conditions can be formalized as:

$$\begin{aligned} p'_i \cdot r'_i &= 0, \ \forall i \in \{1, m\} \\ p'_i + r'_i &> 0, \ \forall i \in \{1, m\} \\ q'_j \cdot t'_j &= 0, \ \forall j \in \{m+1, m+n\} \\ q'_j + t'_j &> 0, \ \forall j \in \{m+1, m+n\} \end{aligned} \tag{D.6}$$

where, it is worth noting that the indexes of p and r are in the same range, and so are those of q and s.

(D.4) and (D.5) both contain $m+n$ variables, but (D.4) only has n equations while (D.5) only has m equations. In order to solve (D.4) and (D.5), it is necessary to eliminate m variables in (D.4) and n variables in (D.5). From (D.6), it is obvious that $m+n$ variables among the total $2m+2n$ variables have to be 0. Therefore, we need to pick m variables in (D.4) to be zero, and pick n variables in (D.5) to be zero. If we know which m variables in (D.4) are 0 and which n variables in (D.5) are 0, we can calculate the nonzero variables for (D.4) and (D.5) separately in the condition that the left equations constitute a non-singular matrix. Since the existence of NE is guaranteed by Nash's Theorem, the problem left for us is how to figure out which m variables in (D.4) and which n variables in (D.5)should be zero. From this aspect, the Lemke-Howson algorithm is an algorithm to find one NE through changing the zero settings little by little until the practical NE emerges.

The concept of *label* can be used to make the next analysis more clear. The labels for (D.4) and (D.5) are denoted as $L(P)$ and $L(Q)$, respectively, which are defined as follows:

$$L(P) = \{i : p'_i = 0, i \in \{1, ..., m\}\} \bigcup \{j : t'_j = 0, j \in \{1 = m+1, ..., m+n\}\} \tag{D.7}$$

$$L(Q) = \{i : r'_i = 0, i \in \{1, ..., m\}\} \bigcup \{j : q'_j = 0, j \in \{1 = m+1, ..., m+n\}\} \tag{D.8}$$

If $L(P)\bigcup L(Q) = \{1, 2, ..., m+n\}$, we will say that the pair of actions (p', q') is *completed labeled*. Apparently, the extraneous solution is completed labeled. When a completed labeled pair, other than extraneous solution, is found and the left two matrices are non-singular, we claim that the problem is solved. To solve the problem by enumeration, we need to first choose a start point, and the start point is completed labeled while the left matrices for the start point are non-singular. Then, for each step, we change the labels for $L(P)$ and $L(Q)$ in turns and guarantee that the left matrices are still non-singular, in other words, can be diagonalized, after the changes. If we get another completed labeled pairs, we find one pair of solutions for (D.4) and (D.5). In the process, the enumeration step is called *pivoting*.

To explain the pivoting, let us consider the following Table D.1.

Table D.1: A sample table M

M	1	2	3	4	=
1	a_1	0	$\boxed{c_1}$	d_1	e_1
2	0	b_2	c_2	d_2	e_2

Apparently, $L(M) = \{3, 4\}$ and the left matrix is non-singular. Next, we want to pivot matrix M on the element of $(1, 3)$, which is c_1. By multiplying c_2/c_1 for row 1 and subtracting the result from row 2 for matrix M, we can get Table D.2.

Table D.2: A sample table M'

M'	1	2	3	4	=
3	a_1	0	c_1	d_1	e_1
2	$-\frac{c_2}{c_1}a_1$	b_2	0	$d_2 - \frac{c_2}{c_1}d_1$	$e_2 - \frac{c_2}{c_1}e_1$

Now, $L(M') = \{1, 4\}$. Through pivoting matrix M on $(1, 3)$, we remove label 3, and add label 1. One requirement for pivoting is that every left matrix in each step should lead to a practical solution, which must be positive. In Table D.2, this requires that $c_1 > 0$ and $c_1/e_1 > c_2/e_2$, which could lead that $e_2 - (c_2/c_1)e_1 > 0$.

The pivoting procedure can be summarized as follows:

```
Data: pivot(M, k0, cl)
Result: M, k, cl
(m,n)=size(M);
k=k0;
max=0;
for i ← 1 to m do
    t = M_{i,k0}/M_{i,n};
    if t > max then
        ind=i;
    end
end
if max > 0 then
    swap(k, cl(ind));
    for i ← 1 to m do
        if i=ind then
            continue;
        end
        for j ← 1 to n do
            M_{i,j} = M_{i,j} − (M_{i,k0}/M_{ind,k0})M_{k,j};
        end
    end
end
return M, k;
```

Algorithm 2: Pivot matrix M for label $k0$

In Algorithm 2, *cl* represent the complementary of the label, which is used to track the variables that can be calculated inside the matrix if the labeled variable are set to zero.

After understanding the algorithm for pivoting, the Lemke-Howson algorithm can be implemented as follows:

In Algorithm 3, the implementation of method *CalculateWithLabel*, which returns the value of p and q basing on the label status, is straightforward and ignored.

Data: LemkeHowson(U_1, U_2)
Result: p, q
(m,n)=size(U_1);
$P = [U_2^T, I, \mathbb{1}]$;
$Q = [I, U_1, \mathbb{1}]$;
LP=[1,...,m];
LQ=[m+1,...,m+n];
k=k0;
while *1* **do**
    Remove(LP,k);
    [P, k]=Pivot(P, k);
    Add(LP,k);
    if *k=k0* **then**
        break;
    end
    Remove(LQ,k);
    [Q, k]=Pivot(Q, k);
    Add(LQ,k);
    if *k=k0* **then**
        break;
    end
end
[p,q]=CalculateWithLabel(P,Q,LP,LQ);
return normal(p), normal(q);

Algorithm 3: The Lemke-Howson algorithm

D.3 An Example

As an example, let us consider the bimatrix game (U_1, U_2) where $U_1 = \begin{pmatrix} 4 & 12 & 8 & 6 \\ 16 & 8 & 12 & 8 \\ 10 & 8 & 10 & 9 \end{pmatrix}$ and $U_2 = \begin{pmatrix} 25 & 5 & 5 & 8 \\ 1 & 15 & 8 & 4 \\ 17 & 10 & 10 & 9 \end{pmatrix}$. Hence, we get matrix P and Q as in Table D.3. Let us first pivot matrix P on label 1 as a start point. Please note that after each pivoting step, we artificially multiply one constant for the corresponding row to make all the elements shown as integer, which will not affect anything but merely for neat display purpose.

Table D.3: The initial matrix P and Q

P	p_1	p_2	p_3	t_4	t_5	t_6	t_7	=
4	$\boxed{25}$	1	17	1	0	0	0	1
5	5	15	10	0	1	0	0	1
6	5	8	13	0	0	1	0	1
7	8	4	9	0	0	0	1	1

L(P)={1, 2, 3}

Q	r_1	r_2	r_3	q_4	q_5	q_6	q_7	=
1	1	0	0	4	12	8	6	1
2	0	1	0	$\boxed{16}$	8	12	8	1
3	0	0	1	10	8	10	9	1

L(Q)={4, 5, 6, 7}

The pivoting for matrix P brings Label 4, then we proceed the pivoting for matrix Q on Label 4, which brings Label 2. These procedures bring us Table D.4.

Table D.4: Matrix P and Q after the first round of pivoting

P	p_1	p_2	p_3	t_4	t_5	t_6	t_7	=
1	25	1	17	1	0	0	0	1
5	0	$\boxed{74}$	33	−1	5	0	0	4
6	0	39	48	−1	0	5	0	4
7	0	92	89	−8	0	0	25	17

L(P)={2, 3, 4}

Q	r_1	r_2	r_3	q_4	q_5	q_6	q_7	=
1	4	−1	0	0	$\boxed{40}$	20	16	3
4	0	1	0	16	8	12	8	1
3	0	−5	8	9	24	20	32	3

L(Q)={2, 5, 6, 7}

For matrix P in Table D.4, due to the duplication of Label 2, we need to pivot matrix P on Label 2, which would bring us Label 5. Then, we pivot matrix Q on Label 5, which brings

us Label 1. This round of pivoting generate Table D.5.

Table D.5: Matrix P and Q after the second round of pivoting

P	p_1	p_2	p_3	t_4	t_5	t_6	t_7	$=$
1	370	0	245	15	−1	0	0	14
2	0	74	33	−1	5	0	0	4
6	0	0	453	−7	−39	74	0	28
7	0	0	3550	−500	−460	0	1850	890

L(P)={3, 4, 5}

Q	r_1	r_2	r_3	q_4	q_5	q_6	q_7	$=$
5	4	−1	0	0	40	20	16	3
4	−32	48	0	640	0	320	192	16
3	−6	−11	20	0	0	20	56	3

L(Q)={1, 2, 6, 7}

After this round of pivoting, we get a completed label, which means a NE has been found. From Table D.5, it is obvious that: $p' = [0.0378,\ 0.0541,\ 0]^T$ and $q' = [0.025,\ 0.075,\ 0,\ 0]^T$. Therefore, $p = normal(p') = [0.4118,\ 0.5882,\ 0]^T$ and $q = normal(q') = q' = [0.25,\ 0.75,\ 0,\ 0]^T$.

D.4 Some Extensions

In Sect. D.2, we assume that matrix U_1 and U_2 are utility matrices. However, the Lemke-Howson algorithm can also be used when U_1 and U_2 are cost matrices after certain simple adjustments. In this section, we shows how to use the Lemke-Howson algorithm to find one NE when the given matrices are cost matrices.

Following the same methodology of (D.1) in Sect. D.2, it is easy to get (D.9) when U_1 and U_2 are cost matrices.

$$\begin{aligned} U_1 \cdot q - r &= v_1 \cdot \mathbb{1} \\ U_2^T \cdot p - t &= v_2 \cdot \mathbb{1} \end{aligned} \tag{D.9}$$

Let us use $\mathbb{M}$ to indicate a matrix, whose dimension decided by the context, and every elements in $\mathbb{M}$ is a relative large number M. Then $(\mathbb{M}-U_1)\cdot q+r = M\cdot\mathbb{1}-U_1\cdot q+r = (M-v_1)\mathbb{1}$. In the same way, $(\mathbb{M}-U_2^T)\cdot p+t = (M-v_2)\mathbb{1}$. Suppose $U_1' = \mathbb{M}-U_1$, $U_2' = \mathbb{M}-U_2$, $v_1' = M-v_1$ and $v_2' = M - v_2$, then we can get (D.10).

$$
\begin{aligned}
U_1' \cdot q + r &= v_1' \cdot \mathbb{1} \\
U_2'^T \cdot p + t &= v_2' \cdot \mathbb{1}
\end{aligned}
\tag{D.10}
$$

Apparently, as long as M is large enough, (D.10) can be solved with the Lemke-Howson algorithm, and the action pair $(p,\ q)$ is also the action pair for U_1 and U_2.

Bibliography

Bibliography

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," *RFC 4271*, 2006.

[2] K. Liu, B. Jabbari, and S. Secci, "Understanding Transit-Edge routing separation: Analysis and characterization," in *Network of the Future (NOF), 2011 International Conference on the*, November 2011, pp. 46 –51.

[3] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)," *RFC 6830*, January 2013.

[4] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. pp. 286–295, 1951.

[5] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," *RFC 3272*, 2002.

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

[7] S. Secci, K. Liu, G. Rao, and B. Jabbari, "Resilient Traffic Engineering in a Transit-Edge Separated Internet Routing," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1 –6.

[8] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Interdomain traffic engineering with BGP," *Communications Magazine, IEEE*, vol. 41, no. 5, pp. 122 – 128, May 2003.

[9] R. Gao, C. Dovrolis, and E. Zegura, "Interdomain Ingress Traffic Engineering Through Optimized AS-Path Prepending," in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, ser. Lecture Notes in Computer Science, R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, and J. Black, Eds. Springer Berlin / Heidelberg, 2005, vol. 3462, pp. 647–658. [Online]. Available: http://dx.doi.org/10.1007/11422778_52

[10] P. Casas, L. Fillatre, and S. Vaton, "Multi Hour Robust Routing and Fast Load Change Detection for Traffic Engineering," in *Communications, 2008. ICC '08. IEEE International Conference on*, May 2008, pp. 5777 –5782.

[11] A. Sridharan and R. Guérin, "Making IGP Routing Robust to Link Failures," in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of*

Computer and Communication Networks; Mobile and Wireless Communications Systems, ser. Lecture Notes in Computer Science, R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, and J. Black, Eds. Springer Berlin / Heidelberg, 2005, vol. 3462, pp. 634–646.

[12] D. Awduche and B. Jabbari, "Internet traffic engineering using multi-protocol label switching (MPLS)," *Computer Networks*, vol. 40, no. 1, pp. 111 – 129, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128602002694

[13] S. Oueslati and J. Roberts, "A new direction for quality of service: flow-aware networking," in *Next Generation Internet Networks, 2005*, April 2005, pp. 226 – 232.

[14] A. Farrel, J. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture," *RFC 4655*, August 2006.

[15] R. Munoz, C. Pinart, R. Martinez, J. Sorribes, G. Junyent, and A. Amrani, "The adrenaline testbed: integrating GMPLS, XML, and SNMP in transparent DWDM networks," *Communications Magazine, IEEE*, vol. 43, no. 8, pp. 40 – 48, August 2005.

[16] R. Zhang and J. Vasseur, "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements," *RFC 4216*, November 2005.

[17] T. Lehman, J. Sobieski, and B. Jabbari, "DRAGON: a framework for service provisioning in heterogeneous grid networks," *Communications Magazine, IEEE*, vol. 44, no. 3, pp. 84 – 90, March 2006.

[18] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 1, pp. 66–66, January 2002. [Online]. Available: http://doi.acm.org/10.1145/510726.510740

[19] "Configuring BGP to Select Multiple BGP Paths," *JUNOS documentation*. [Online]. Available: http://www.juniper.net/techpubs/software/junos/junos94/swconfig-routing/configuring-bgp-to-select-multiple-bgp-paths.html

[20] "BGP Best Path Selection Algorithm," *Cisco documentation*. [Online]. Available: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094431.shtml#bgpmpath

[21] E. Elena, J. Rougier, and S. Secci, "Characterisation of AS-level path deviations and multipath in Internet routing," in *Next Generation Internet (NGI), 2010 6th EURO-NF Conference on*, June 2010, pp. 1 –7.

[22] S. Secci, J.-L. Rougier, A. Pattavina, F. Patrone, and G. Maier, "PEMP: Peering Equilibrium MultiPath Routing," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, November 30-December 4 2009, pp. 1 –7.

[23] W. Xu and J. Rexford, "MIRO: multi-path interdomain routing," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 171–182, August 2006. [Online]. Available: http://doi.acm.org/10.1145/1151659.1159934

[24] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural Guidelines for Multipath TCP Development," *RFC 6182*, March 2011.

[25] A. Orda, R. Rom, and N. Shimkin, "Competitive routing in multiuser communication networks," *IEEE/ACM Trans. Netw.*, vol. 1, no. 5, pp. 510–521, October 1993. [Online]. Available: http://dx.doi.org/10.1109/90.251910

[26] E. Altman and H. Kameda, "Equilibria for multiclass routing in multi-agent networks," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 1, 2001, pp. 604 –609 vol.1.

[27] R. W. Rosenthal, "A class of games possessing pure-strategy Nash equilibria," *International Journal of Game Theory*, vol. 2, pp. 65–67, 1973, 10.1007/BF01737559. [Online]. Available: http://dx.doi.org/10.1007/BF01737559

[28] A. Lazar, A. Orda, and D. Pendarakis, "Virtual path bandwidth allocation in multi-user networks," in *INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE*, April 1995, pp. 312 –320 vol.1.

[29] M. Hong and L. Yang, "MMC03-5: A Game Theoretic Approach to Multi-Stream QoS Routing," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, November 2006, pp. 1–5.

[30] T. Roughgarden and É. Tardos, "How bad is selfish routing?" *Journal of the ACM (JACM)*, vol. 49, no. 2, pp. 236–259, 2002.

[31] T. Roughgarden, "The price of anarchy is independent of the network topology," *Journal of Computer and System Sciences*, vol. 67, no. 2, pp. 341–364, 2003.

[32] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[33] M. Patriksson, *The traffic assignment problem : models and methods.* Utrecht, The Netherlands, 1994.

[34] Y. Masuda and S. Whang, "Capacity Management in Decentralized Networks," *Management Science*, vol. 48, no. 12, pp. 1628–1634, 2002.

[35] M. T. Hsiao and A. A. Lazar, "Optimal decentralized flow control of Markovian queueing networks with multiple controllers," *Performance Evaluation*, vol. 13, no. 3, pp. 181 – 204, 1991.

[36] Y. A. Korilis and A. A. Lazar, "On the existence of equilibria in noncooperative optimal flow control," *J. ACM*, vol. 42, no. 3, pp. 584–613, May 1995.

[37] S. Secci, J.-L. Rougier, A. Pattavina, F. Patrone, and G. Maier, "Peering Equilibrium Multipath Routing: A Game Theory Framework for Internet Peering Settlements," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 2, pp. 419 –432, April 2011.

[38] S. Secci, H. Ma, B. Helvik, and J.-L. Rougier, "Resilient Inter-Carrier Traffic Engineering for Internet Peering Interconnections," *Network and Service Management, IEEE Transactions on*, vol. 8, no. 4, pp. 274 –284, December 2011.

[39] R. La and V. Anantharam, "Optimal routing control: repeated game approach," *Automatic Control, IEEE Transactions on*, vol. 47, no. 3, pp. 437 –450, March 2002.

[40] D. Saucez, B. Donnet, L. Iannone, and O. Bonaventure, "Interdomain traffic engineering in a locator/identifier separation context," in *Internet Network Management Workshop, 2008. INM 2008. IEEE*, October 2008, pp. 1 –6.

[41] D. Monderer and L. S. Shapley, "Potential Games," *Games and Economic Behavior*, vol. 14, no. 1, pp. 124 –143, May 1996.

[42] D. Meyer, "University of oregon route views archive project," *at http://archive.routeviews.org*.

[43] Details and codes, *at http://cnl.gmu.edu/TAVRI/research/*.

[44] R. Radner, "Collusive behavior in noncooperative epsilon-equilibria of oligopolies with long but finite lives," *Journal of Economic Theory*, vol. 22, no. 2, pp. 136–154, 1980.

[45] R. McKelvey and A. McLennan, "Computation of equilibria in finite games," *Handbook of computational economics*, vol. 1, pp. 87–142, 1996.

[46] C. Lemke and J. Howson Jr, "Equilibrium points of bimatrix games," *Journal of the Society for Industrial & Applied Mathematics*, vol. 12, no. 2, pp. 413–423, 1964.

[47] N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[48] L. S. Shapley, "A note on the Lemke-Howson algorithm," in *Pivoting and Extension*, ser. Mathematical Programming Studies. Springer Berlin Heidelberg, 1974, vol. 1, pp. 175–189.

[49] K. Kompella, J. Drake, S. Amante, W. Henderickx, and L. Yong, "The Use of Entropy Labels in MPLS Forwarding," *draft-ietf-mpls-entropy-label-05*, August 2012.

[50] "Cisco Advanced Virtual Private LAN Service: Unite Geographically Dispersed Data Centers," *Cisco documentation*. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns949/ns304/ns975/product_bulletin_c25-602184.pdf

[51] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," *RFC 4984*, 2007.

[52] L. Cittadini, W. Mü andhlbauer, S. Uhlig, R. Bush, P. Franç andois, and O. Maennel, "Evolution of Internet Address Space Deaggregation: Myths and Reality," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 8, pp. 1238 –1249, october 2010.

[53] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "BGP Churn Evolution: a Perspective from the Core," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.

[54] ——, "On the scalability of BGP: the role of topology growth," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 8, pp. 1250–1261, 2010.

[55] A. Feldmann, "Internet clean-slate design: what and why?" *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 59–64, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1273445.1273453

[56] E. Nordmark and M. Bagnulo, "Shim6: Level 3 multihoming shim protocol for IPv6," *RFC 5533*, 2009.

[57] R. Moskowitz and P. Nikander, "Host identity protocol (HIP) architecture," *RFC 4423*, 2006.

[58] Y. Wang, J. Bi, and J. Wu, "Empirical analysis of core-edge separation by decomposing Internet topology graph," in *Proc. of IEEE GLOBECOM*, 2010.

[59] A. Dhamdhere and C. Dovrolis, "Ten years in the evolution of the internet ecosystem," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008, pp. 183–196. [Online]. Available: http://doi.acm.org/10.1145/1452520.1452543

[60] Z. Mao, L. Qiu, J. Wang, and Y. Zhang, "On AS-level path inference," in *Proc. of ACM SIGMETRICS*, 2005, pp. 339–349.

[61] D. Cohen, "All the world's a net," *New Scientist*, vol. 174, no. 2338, pp. 24–29, 2002.

[62] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social Networks*, 2010.

[63] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang, "IPv4 address allocation and the BGP routing table evolution," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 1, pp. 71–80, 2005.

[64] R. Gagliano, E. Grampin, J. Baliosian, X. Masip-Bruin, and M. Yannuzzi, "Understanding IPv4 prefix de-aggregation: challenges for routing scalability," in *Integrated Network Management-Workshops, 2009. IM '09. IFIP/IEEE International Symposium on*, 2009, pp. 107–112.

[65] C. Labovitz, G. Malan, and F. Jahanian, "Internet routing instability," *Networking, IEEE/ACM Transactions on*, vol. 6, no. 5, pp. 515–528, 1998.

[66] R. W. Rosenthal, "A class of games possessing pure-strategy Nash equilibria," *International Journal of Game Theory*, vol. 2, pp. 65–67, 1973, 10.1007/BF01737559. [Online]. Available: http://dx.doi.org/10.1007/BF01737559

Curriculum Vitae

Kunpeng Liu grew up in China. He attended Tsinghua University, Beijing, China, where he received his Bachelor of Engineering in Electrical Engineering in 2005. He worked for Lucent Technologies, China as a Senior Technical Associate since then till 2008. He came to U.S. in 2008, and received his Ph.D. in Electrical and Computer Engineering in 2013.