On the Interactions of Overlay Routing

by

Gene Moo Lee, B.S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Arts

The University of Texas at Austin

May 2006

On the Interactions of Overlay Routing

Approved by Supervising Committee:

Yin Zhang, Supervisor

Aloysius K. Mok

To my family

Acknowledgments

First of all, I want to thank my Lord, Jesus Christ, to love me and bless my life. Without your blessing, nothing is possible.

I want to thank Prof. Yin Zhang for his great advice with this work. He gave me such a great direction on my research. I will never forget the moment of my first publication with him. Thanks also to Prof. Aloysius K. Mok for being in the thesis committee and supporting my studies at UT.

My studies would be impossible without the inspiration of great scholars around me. First, I would like to thank Prof. Vladimir Lifschitz. His enthusiasm as a logician and mathematician inspired me a lot. Thanks also to Prof. Dale Stahl and Prof. David Kendrick at Department of Economics, for introducing me the game theory and computational aspects of economics. I also want to thank my advisors at Korea University: Prof. Jin-Young Choi, Prof. Youn-seo Choi, and Prof. Hoh Peter In.

I am blessed to share the spiritual life with the fellows at Korean Baptist Church of Austin. First, I want to thank Chanje Lee for guiding my life as a Christian. Thanks also to Mr. Yong-Min Wang, Pastor Sung Bae Kim, and all the SALT members. Moreover, I was fortunate to meet all the choir members at our church. I really enjoyed the moment to praise the Lord with them every week.

I was also fortunate to meet wonderful colleagues at UT. Thanks to my room-mate Yonghyun Kim, my racquetball teacher TaeWon Cho, Han Hee Song, Byeongcheol (BK) Lee, Jungwoo Ha, Taehwan Choi, Doo Soon Kim, Dmitry Kit, Joohyung Lee, Ajay

Mahimkar, Honguk Woo, and Mahir Yildirim. Without them, my life at Austin would be really dry. Thanks, guys.

I can never forget the moment I shared with my friends in Korea. Thanks to my indeed friends: the movie maker Tae-Ho Kang, the activist Doo-Hyoung Nam, Byung-Jae Kang, Junkwan Park, Jee-Ho Park, Sun-Jun Kim, Kie-Yong Shin, Hyun-Gu Kim, Suny-oung Yeom and other friends from TFLHS. Thanks also to fellows from Korea University: Tae-bong Sun, Meesun Yu, JeongYun Lim, ByeongChul Lee, Seung-Taek Yu, and other Linuxers.

Last but not least, I'd like to thank my family. Deepest thanks to my father Dr. Sang-Won Lee for his continuous support and encouragement throughout this work. His whole life is the inspiration to me. I also thank my new mother Mrs. Ok-Hyun Kim for her care, my sister Ga-Won Lee (I'm proud you become a shepherd), and my two brothers Sung-Hwan Kim and Hyung-Jun Kim. Lastly, I give my love to my mother the late Mrs. Kyung-Ja Hwang, who is now in Heaven. This thesis is dedicated to her.

GENE MOO LEE

The University of Texas at Austin
May 2006

On the Interactions of Overlay Routing

Gene Moo Lee, M.A.
The University of Texas at Austin, 2006

Supervisor: Yin Zhang

Overlay routing has been successful as an incremental method to improve the current Internet routing by allowing users to select the Internet paths by themselves. By its nature, overlay routing has selfish behavior, which makes impact on the related components of the Internet routing. In this thesis, we study three interactions related to overlay routing. First, overlay routing changes the traffic patterns observed by the network operating side, which uses traffic engineering techniques to cope with the dynamic traffic demands. We improve this *vertical interaction* between overlay routing and traffic engineering. Secondly, the performance of overlay routing may be affected by the action of other coexisting overlays. An initial result on the *horizontal interaction* among multiple overlays is given. Lastly, within a single overlay network, overlay nodes can be regarded as independent decision makers, who act strategically to maximize individual gain. We design an incentive-based framework to achieve Pareto-optimality in the *internal interaction* of overlay routing.

Contents

List of Figures

Chapter 1

Introduction

The Internet is organized as multiple autonomous systems peering each other, and these systems are operated by independent Internet Service Providers (ISPs), which seek to maximize the profit. The Internet paths between ISPs are mainly depending on the business relationship. Thus, the end-to-end path performance may not be the main concern for inter-domain routing, and the default Internet routing may not be optimized in the end user's point of view. Previous studies [34, 38] have shown that there is inherit inefficiency in network-level routing.

In addition, today's Internet only provides a best-effort service. In other words, the Internet does not guarantee some quality of service (QoS) requirements. For the past decades, there have been many efforts to provide QoS in the Internet, such as IntServ [11] and DiffServ [25]. However, those methods have ultimate challenges: the whole IP infrastructure needs to be changed for such mechanism to be deployed.

1.1 Overlay Routing

Overlay routing has been proposed as an *incremental* method to enhance the current Internet routing without requiring additional functionality from the IP routers. Overlay techniques

have been successful for many applications, including application-layer multicast [14, 21, 37], web content distribution [1], and overlay routing [8, 34].

In an overlay network, several overlay nodes form an application-layer logical network on top of the IP layer network. Overlay networks enable users to make routing decisions at the application layer by relaying traffic among overlay nodes. We can achieve better route than default IP routing because some problematic and slow links can be bypassed. In addition, overlay routing can take advantage of some fast and reliable paths, which could not be used in the default IP routing due to the business relationship.

1.2 Interactions of Overlay Routing

By its nature, overlay routing has selfish behavior. In other words, overlay acts strategically to optimize its performance. This nature of overlay makes impact on the related components of the network. In this thesis, we study various interactions involved with overlay routing and its related components.

1.2.1 Vertical Interaction

First, overlay routing has *vertical interaction* with IP layer's traffic engineering. Whenever overlay network changes its logical routing, the physical traffic pattern changes, which is observed by the underlay routing. Network operators use traffic engineering techniques [5, 33, 36] to adapt the routing to cope with the new traffic demands. This new routing, in turn, changes the link latency observed by the overlay network, and then overlay makes another decision to change its routing.

Network-layer routing protocols care about the network as a whole, in order to provide better service to all the users. However, the main objective of overlay routing is to minimize its own traffic latency. Then an interesting issue is to understand the interaction between overlay routing and IP routing.

The interaction between overlay routing and traffic engineering was first addressed

by Qiu et al. [32], where the authors investigate the interaction of overlay routing with OSPF and MPLS traffic engineering. Keralapura et al. [24] examine the interaction dynamics between the two layers of control from an ISP's view. Liu et al. [27] formulate the interaction as a two-player game, where overlay attempts to minimize its delay and traffic engineering tries to minimize the network cost. The paper shows that the interaction causes a severe oscillation problem to each player and that both players gain little or nothing as the interaction proceeds.

In this thesis, we propose *TE-aware overlay routing*, which takes the objective of underlay routing into account, instead of blindly optimizing its performance. Moreover, we argue that it is better off for both players if the underlay routing is oblivious to the traffic demands. We suggest COPE [39] as a strong candidate for this purpose.

1.2.2 Horizontal Interaction

The second interaction involved with overlay routing is called *horizontal interaction*. Given that the overlay-based techniques widely get deployed, multiple overlay networks may coexist on top of a given underlay network.

Qiu et al. [32] first address interactions between coexisting overlays, where the authors investigate the performance of selfish routing after the system reaches the Nash equilibrium point. Seshadri and Katz [35] investigate the performance of greedy route selection in a variety of scenarios. Their finding is that overlay routing can perform poorly if the overlay flows comprise a significant fraction of link capacities. Jiang et al. [22] propose *overlay optimal routing* and compare the performance with selfish routing. The authors prove the existence of the Nash equilibrium in the multiple overlay game and show that the game may not be Pareto optimal in some scenarios. Keralapura et al. [23] describe how multiple similar or dissimilar overlay networks could experience race conditions, resulting oscillations in route selection and network load.

Following the related works, we formulate the horizontal interaction as a non-

cooperative game among overlays, where each overlay network makes simultaneous moves based on the observation of the current network status. In addition, we include underlay routing as another player of the game and analyze the interaction process.

1.2.3 Internal Interaction

Lastly, within a single overlay network, overlay nodes have *internal interaction* with each other. Instead of considering the behavior of overlay network as a whole, overlay nodes are regarded as independent decision-makers to optimize their own performance. In this model, individual overlay node selectively forwards other's traffic to maximize its own benefit and to minimize the cost to relay transit traffic.

There have been extensive research [10, 12, 20, 26, 30] to design incentives to cooperate in Peer-to-Peer (P2P) networks. The incentive-based systems in P2P file sharing have fundamental difference from that of overlay routing: players in overlay routing are well-identified routers, but P2P users can easily hide their identity to be anonymous players. Thus, the framework for overlay routing must capture the repeated nature of interactions among overlays.

In this thesis, we view the transit traffic forwarding in overlay routing as a non-cooperative two-player repeated game. We propose an incentive-based framework to stimulate the cooperation of overlay nodes. We show the feasibility of our proposed system by analytic proofs and simulation results.

1.3 Thesis Overview

The thesis is organized as follows. In Chapter 2, we discuss the vertical interactions between overlay routing and underlay routing. Horizontal interactions among multiple overlays are studied in Chapter 3. In Chapter 4, an incentive-based framework for internal interaction is proposed and evaluated. We conclude the thesis and discuss future direction in Chapter 5.

Chapter 2

Improving the Interaction between Overlay Routing and Traffic Engineering

2.1 Introduction

In every networked system, there are two different standpoints involved with it: service providers who operate the system and individual users who take advantage of the service. For example, in transportation system, the transportation authority operates the traffic signal to make efficient traffic flow and to avoid traffic jam. On the other side, the drivers make their own decisions on the directions to the destination. They try to bypass some congested roads to arrive their destination as soon as possible.

The authority and drivers have different viewpoints of the transportation. The operation side has a global view of the transportation system and wants to provide good service to all people, whereas individual drivers have just local view of the roads and the priority is their own driving. We have similar situation in the Internet routing. The operating side, Internet Service Providers, use traffic engineering techniques to adapt the routes to make

better service to all the customers. Network users together form an overlay network and make their own routing decisions to achieve better Internet routes.

In an overlay network, several overlay nodes form an application-layer logical network on top of the IP layer network. Overlay network enables users to control their own Internet paths by relaying traffic through logical links between overlay nodes. We can achieve better route than default IP routing because some problematic and slow links can be bypassed. Plus, we can use some fast and reliable paths, which could not be used with the default routing due to the business relationship or policy issues.

By the underlying definition of overlay routing, it has selfish behavior. In other words, users just want to get their best performance, regardless of their impact on the whole network. However, the logical path between two overlay nodes is governed by the underlay routing protocol, such as OSPF [5], MPLS [33], or BGP [36], which care about the network as a whole. Then an interesting issue is to understand the interaction between overlay routing and underlay routing. We term this as *vertical interaction* of overlay routing.

Qiu et al. [32] shows that the adaptive nature of selfish overlays can significantly reduce the effectiveness of traffic engineering by making network traffic less predictable. Liu et al. [27] formalizes this problem as a non-cooperative two-player game, and shows the actual evidence where this misalignment of optimizations makes inefficiency.

In this chapter, we suggest methods to improve the interaction between overlay routing and traffic engineering. The basic idea is to modify the objectives of each side to take other party's objectives into consideration. First, we make the underlay routing to be oblivious to the underlay traffic demand so that the change of overlay traffic does not hurt the performance of traffic engineering. We find COPE [39] as a strong candidate for this purpose. For the overlay part, we limit the selfishness of the overlay routing to make sure the behaviors of overlay do not conflict with the objective of underlay routing. We term that as *TE-aware overlay routing*. With extensive simulation, we show how the vertical interaction is improved by the proposed methods.

$(i,j), l$	physical link
(i',j')	logical link
$cap(l)$	capacity of a physical link l
$v_{st}(l)$	flow of d_{st} on link l
$f_{st}(l)$	fraction of d_{st} on link l
$t(l)$	traffic rate at link l
d_{st}	total TE demand on physical node pair (s,t)
$d_{s't'}$	overlay demand on pair (s',t')
d_{st}^{under}	TE demand due to underlay traffic
$d_{st}^{overlay}$	TE demand due to overlay flow
$P^{(s't')}$	set of logical paths from s' to t'
$\delta_p^{s't'}$	path mapping coefficient
$h_p^{(s't')}$	overlay flow on logical path p

Table 2.1: Notations for vertical interaction

The remainder of the chapter is as follows. We formally describe underlying model in Section 2.2. In Section 2.3, we formulate the interaction of overlay routing and traffic engineering as a non-cooperative two-player game. Then various underlay routing schemes are described in Section 2.4, and selfish overlay routing and TE-aware overlay routing are given in Section 2.5. Section 2.6 briefly explains how the implementation is done, and Section 2.7 evaluates the proposed methods with simulation. Lastly, we conclude the chapter and discuss future direction in Section 2.8.

2.2 Model

In this section, we describe the mathematical model, which will be used throughout the chapter. Basically, traffic engineering and overlay have different viewpoints of the network. Network operators know all the underlying structure of the physical network, whereas overlay has a logical view of the network.

Table 2.1 summarizes the notations for vertical interaction. First, we use $G = (V, E)$ to denote an underlay network, where V is the set of physical nodes and E is the set

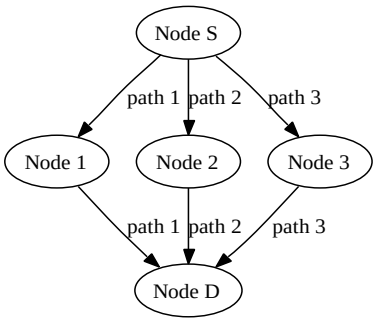

Figure 2.1: An overlay network with five nodes: there are three possible logical paths for the logical demand from s to d.

of edges between nodes. We use l or (i, j) to denote a link and $cap(l)$ to refer the capacity of link l. For the virtual network of overlay, we use $G' = (V', E')$. In G', we use i' to represent the overlay node built upon physical node i in underlay graph G. Overlay node i' is connected to j' by a *logical link* (i', j'), which corresponds to a physical *path* from i to j in G.

Now, we need to have different notations for overlay and underlay traffic demands: d_{st} is used to indicate the total traffic demand from node s to t, including overlay and non-overlay traffics, and d_{st} is a sum of d_{st}^{under} and $d_{st}^{overlay}$. d_{st}^{under} refers to the background traffic by non-overlay demands. Next, it is important to differentiate $d_{st}^{overlay}$ from $d_{s't'}$: $d_{s't'}$ indicates the logical traffic demand from overlay node s' to t', whereas $d_{st}^{overlay}$ is the physical traffic demand on physical node pair (s, t), generated by overlay network. In other words, $d_{st}^{overlay}$ is computed by the overlay routing based on the current logical demand $\{d_{s't'} | \forall s', t' \in E'\}$.

The third group of notations is for the overlay routing. $P^{(s't')}$ is the set of logical paths from s' to t'. $\delta_p^{s't'}$ is the path mapping coefficient, where the value is 1, if logical link (s', t') is on logical path p, and 0, otherwise. $h_p^{(s't')}$ is the amount of overlay demand $d(s't')$ flowing on logical path p. Let us illustrate it by an example. In Figure 2.1, five nodes form an overlay network. Then $P^{(s,d)} = \{s \to 1 \to d, s \to 2 \to d, s \to 3 \to d\}$ and let

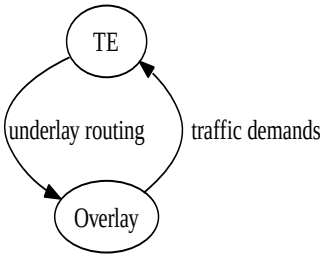

Figure 2.2: vertical interaction game: TE determines the physical routing, which decides link latency experienced by overlay. Given the observed latency, overlay optimizes its logical routing and changes the physical traffic demands, which, in turn, affects the underlay routing.

those paths to be p_1, p_2, p_3, respectively. Then δ values will be as follows: $\delta_1^{s1} = 1, \delta_1^{1d} = 1, \delta_2^{s2} = 1, \delta_1^{2d} = 1$, and so on. Say $d_{s't'} = 3$ and the overlay traffic is equally splitted to the three logical paths, then $h_1^{s't'} = h_2^{s't'} = h_3^{s't'} = 1$.

2.3 Vertical Interaction Game

Based on the formulations in the previous section, traffic engineering and overlay routing are coupled through the mapping from the logical level path to physical level links. We can formulate the interaction as a non-cooperative two-player game as described in Figure 2.2.

The first player is the ISP's traffic engineering and the second player is the overlay routing for the user's side. The interaction consists of *sequential* moves of the two players. Each player takes turn and makes action to optimize its performance. Based on the overlay demand and flow conditions on the physical links, overlay calculates the optimal flows on the logical routing. These logical flows and the underlay background traffic are coupled to form the total traffic matrix, which is the input for traffic engineering. Then traffic engineering optimizes its performance by adapting the flows on the physical links, which in turn affects the delays experienced by the overlay. This interaction continues until the two players come up with the Nash equilibrium point [17, 29].

In game theory, the Nash equilibrium (named after John Nash, who proposed it) is a

kind of optimal collective strategy in a game involving two or more players, where no player has anything to gain by changing only his or her own strategy. If each player has chosen a strategy and no player can benefit by changing his or her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium.

Liu et al. [27] prove the existence of Nash Equilibrium in a simple interaction game, where the topology consists of three nodes and there is a single demand between two nodes. Even though we might prove the existence of a convergent point, the interaction process does not guarantee that two players' behaviors converge to the Nash equilibrium. Moreover, if the game gets complicated, it is even harder to anticipate the interaction process. The authors show that in a realistic scenario, both traffic engineering and overlay routing experience substantial performance loss due to the oscillation.

The main direction of our work is to improve the vertical interaction between overlay routing and traffic engineering. First, we want the interaction game to converge faster because the oscillation in this game degrades the performance of both players. Next, we try to reduce the performance variation in the transient oscillation process.

2.4 Traffic Engineering

In this section, we now formulate three traffic engineering schemes: Multi-Protocol Label Switching (MPLS) [33], oblivious routing [9], and Convex-hull-based Optimal traffic engineering with Penalty Envelope (COPE) [39]. There are other IP routing protocols not considered in the thesis, such as Open Shortest Path First (OSPF) [5]. We do not consider this method because it is shown in [32] the vertical interaction of the scheme is inefficient.

The output of traffic engineering is IP-layer routing, which specifies how traffic of each Origin-Destination (OD) pair is routed across the network. Typically, there is path diversity, that is, there are multiple paths for each OD pair, and each path routes a fraction of the traffic.

2.4.1 Multi-Protocol Label Switching Traffic Engineering

Multi-Protocol Label Switching (MPLS) [33] provides an efficient support of explicit routing, which is the basic mechanism for traffic engineering. Explicit routing allows a particular packet stream to follow a predetermined path rather than a path computed by hop-by-hop destination based routing such as OSPF or IS-IS.

The combination of MPLS technology and its traffic engineering capabilities enable the network operator to adaptively load-balance the traffic demands to optimize the network performance. There are two possible ways to describe the network performance: maximum link utilization and total link latency.

The first possible performance metric is maximum link utilization. Network operators sometimes worry about over-loaded links, because these links can be a bottleneck for the whole network performance. A slight change of the traffic pattern may overload the over-utilized links. Therefore, we want to minimize the maximum link utilization.

Given the traffic demand matrix $\{d_{st}|\forall s,t \in V\}$, the goal of MPLS traffic engineering is to choose a physical link flow allocation $\{f_{st}(l)|\forall s,t \in V, \forall l \in E\}$ which minimizes the maximum link utilization. The Linear Program model is given as follows:

$$\begin{aligned} \min \quad & r \\ \text{subject to} \quad & f_{st}(l) \text{ is a routing} \\ & \forall \text{ link } l : \sum_{s,t} f_{st}(l) d_{st} / cap(l) \leq r \end{aligned}$$

Here, the first constraint ensures that the given routing satisfies the flow conservation constraints. Basically, for each router, total in-coming traffic should be equal to total out-going

traffic. It can be described as the following equations:

$$\sum_{l|dst(l)=y} f_{st}(l) - \sum_{l|src(l)=y} f_{st}(l) = \begin{cases} 1 & \text{if } y = t, \\ -1 & \text{if } y = s, \\ 0 & \text{otherwise} \end{cases}$$

for each OD pair s, t. Here, $l|dst(l) = y$ indicates all links destined to y, and $l|src(l) = y$ means all links sourced from y.

For the next option, we can use total link latency as the network performance metric. It is clear that faster the link speed is better, and we want to minimize the total link latency throughout the network. We use the *M/M/1* delay formula to calculate link cost. For a physical link l with capacity $cap(l)$, if its traffic rate is $t(l)$, the total delay experienced by traffic engineering on the links is $\frac{t(l)}{cap(l)-t(l)}$.

Given the traffic demand matrix $\{d_{st}|\forall s, t \in V\}$, the goal of traffic engineering is to choose a physical link flow allocation $\{f_{st}(l)|\forall s, t \in V, \forall l \in E\}$ that minimizes network costs:

$$\begin{aligned} \min \quad & \sum_l \frac{t(l)}{cap(l) - t(l)} \\ \text{subject to} \quad & f_{st}(l) \text{ is a routing} \\ & \forall \text{ link } l : t(l) = \sum_{s,t} f_{st}(l) d_{st} \end{aligned}$$

Note that the link latency function is non-linear, which makes the optimization process to be time-consuming. According to [18, 19], the cost of a link can be modeled with a piecewise-linear, increasing, convex function with slopes as follows:

$$u_l(x/cap) = \begin{cases} 1 & : x/cap \in [0, 1/3) \\ 3 & : x/cap \in [1/3, 2/3) \\ 10 & : x/cap \in [2/3, 9/10) \\ 70 & : x/cap \in [9/10, 1) \\ 500 & : x/cap \in [1, 11/10) \\ 5000 & : x/cap \in [11/10, \infty) \end{cases}$$

where x is the load on link l with capacity cap. We will use this linear function to calculate the latency for overlay routing in Section 2.5.

2.4.2 Oblivious Routing

One of the important components of traffic engineering is to understand the traffic flow. Previously discussed MPLS traffic engineering optimizes the paths based on the currently observed traffic matrix. Unfortunately, measuring and predicting traffic demands are really difficult problems. Flow measurements are rarely available on all links and Ingress/Egress points. Moreover, demands change over time on special events like DoS attack and flash crowds, or failures internal or external to the network. It seems that the most one can hope is some approximate picture of demands, not necessarily the very current one.

Oblivious routing [9] is proposed to resolve this issue. It calculates an optimal routing which performs reasonably well *independently* of traffic demands. In other words, this "demand oblivious" routing is designed with little knowledge of the traffic matrix (TM), taking only the topology along with link capacities into account. Oblivious routing can be

computed by the following Linear Program model:

$$
\begin{aligned}
\min \quad & r \\
\text{subject to} \quad & f_{st}(l) \text{ is a routing} \\
& \forall \text{ link } l : \sum_{m} cap(m)\pi(l,m) \leq r \\
& \forall \text{ link } l, \forall \text{ pair } s \rightarrow t : f_{st}(l)/cap(l) \leq p_l(s,t) \\
& \forall \text{ link } l, \forall \text{ node } s, \forall \text{ edge } e = t \rightarrow v : \pi(l, \text{link-of}(e)) + p_l(s,t) - p_l(s,v) \geq 0 \\
& \forall \text{ link } l, m : \pi(l,m) \geq 0 \\
& \forall \text{ link } l, \forall \text{ node } s : p_l(s,s) = 0 \\
& \forall \text{ link } l, \forall \text{ node } s,t : p_l(s,t) \geq 0
\end{aligned}
$$

Note that the model above does not include any information about the traffic demand. The output routing of the optimization is the optimal solution for all possible traffic matrices. In the case we have some knowledge of traffic demand such as lower bounds and upper bounds, we can produce an optimal routing for the specific range traffic matrices. Interesting authors may refer to [9] for details.

2.4.3 Convex-Hull-Based Optimal Traffic Engineering with Penalty Envelope

MPLS Traffic Engineering can be regarded as an extreme case of online adaptation. An advantage of this scheme is that it achieves the best performance for the current traffic demand. However, if there are significantly fast traffic changes, such method can suffer a large transient penalty. Oblivious routing is a way to handle unpredicted traffic spikes. However, a potential drawback of completely oblivious routing is its sub-optimal performance for the normal traffic demand.

Convex-hull-based traffic engineering with penalty envelope (COPE) [39] is proposed as a hybrid combination of predication-based optimal routing and oblivious routing. COPE handles both dynamic traffic and dynamic inter-domain routes and, at the same time,

achieves close-to-optimal performance for normal, predicted traffic matrices.

The optimization mainly consists of two parts: convex-hull-based optimal traffic engineering and penalty envelope constraints. Let *D* be the convex hull of the set of traffic matrices $\{D_1, ..., D_H\}$. Then the problem of finding optimal min-max ratio of the network on the set of traffic matrices *D* can be formulated as following Linear Program:

$$\begin{aligned} \min \quad & r \\ \text{subject to} \quad & f_{st}(l) \text{ is a routing} \\ & \forall \text{ link } l, \forall \text{ TM } D = \sum_{h=1}^{H} \alpha_h D_h, \alpha_h \geq 0, OU(D) = 1 : \\ & \sum_{st} d_{st} f_{st}(l)/cap(l) \leq r \end{aligned}$$

Next, the penalty envelope constraint restricts that the routing *f* has maximum performance ratio less than or equal to $\overline{r}$. This can be formalized as following set of linear constraints:

$$\forall \text{ link } l : \sum_{m} cap(m)\overline{\pi}(l,m) \leq \overline{r}$$

$$\forall \text{ link } l, \forall \text{ pair } s \rightarrow t : f_{st}(l)/cap(l) \leq \overline{p}_l(s,t)$$

$$\forall \text{ link } l, \forall \text{ node } s, \forall \text{ edge } e = t \rightarrow v : \overline{\pi}(l, \text{link-of}(e)) + \overline{p}_l(s,t) - \overline{p}_l(s,v) \geq 0$$

$$\forall \text{ link } l, m : \overline{\pi}(l,m) \geq 0$$

$$\forall \text{ link } l, \forall \text{ node } s : \overline{p}_l(s,s) = 0$$

$$\forall \text{ link } l, \forall \text{ node } s, t : \overline{p}_l(s,t) \geq 0$$

The convex-hull-based Linear Program takes the input as a set of possible traffic matrices. In our simulation, however, we use a single currently observed traffic matrix. Still, COPE is shown to make an excellent performance for the dynamic change of the traffic patterns.

2.5 Overlay Routing

In this section, we formulate the objective functions for overlay routing. We start with the default overlay routing, which we term *selfish overlay routing*. Given the current underlay routing and experienced link latency, selfish overlay tries to minimize its total latency by changing the loads for each logical path in the overlay network.

Then, we propose a variation of overlay routing. Basically, we include additional constraints to the original overlay optimization so that overlay takes the presence of traffic engineering into account. We term this as *TE-aware overlay routing*.

2.5.1 Selfish Overlay Routing

The overlay routing algorithm determines a logical path flow allocation $\{h_p^{s't'} | \forall s', t' \in V', \forall p \in P^{(s't')}\}$ that minimizes the average delay experienced by the overlay users, whereas traffic engineering determines the physical flow. By $h_p^{s't'}$, we denote the logical overlay demand from s' to t' allocated to path p.

Individual overlay users may choose their routes independently by probing the underlay network. However, we assume that a centralized entity calculates routes for all overlay users. Given the physical network topology, underlay routing, and experienced latency for each link, optimal overlay routing can be obtained by solving the following non-linear optimization problem:

$$
\begin{aligned}
\min \quad & \sum_l \frac{t(l)^{overlay}}{cap(l) - t(l)} \\
\text{subject to} \quad & h_p^{s't'} \text{ is a logical routing} \\
& \forall \text{ link } l : t(l) = \sum_{s,t} f_{st}(l)(d_{st}^{under} + d_{st}^{overlay}) \\
& \forall \text{ link } l : t(l)^{overlay} = \sum_{s,t} d_{st}^{overlay} f_{st}(l) \\
& \forall s, t \in V : d_{st}^{overlay} = \sum_{s',t',p} \delta_p^{s't'} h_p^{(s't')}
\end{aligned}
$$

```
TE-aware-overlay-routing (latency) {
  if(latency < avg_latency) load-balancer(latency);
  else limited-optimizer(max-load);
}
```

Figure 2.3: TE-aware Overlay Routing algorithm

The first constraint ensures that the logical routing satisfies the logical flow conservation constraints. This can be expressed as follows:

$$\forall s', t' \in V' : \sum_{p \in P^{(s't')}} h_p^{(s't')} = d^{(s't')}$$
$$\forall s', t' \in V' : h_p^{(s't')} \geq 0.$$

Note that the main objective of problem is non-linear. But we can again linearize the non-linear part of the program by using the same technique used for the traffic engineering optimization. Refer to the Section 2.4 for details.

2.5.2 TE-Aware Overlay Routing

Based on the selfish overlay routing, we can include additional constraints to ensure the overlay is *TE-aware*. By TE-awareness, we mean the selfishness of the overlay is limited by some bound so that the action of overlay does not offensively affect the traffic engineering's optimization process.

The basic idea is this: (1) when the current latency is below the average latency, the overlay tries to minimize its own traffic amount, given that the current latency is preserved (load-balancer). (2) If the latency is above the average, then overlay changes the logical routing to improve the latency, but, at the same time, it avoids a specific link to overloaded (limited-optimizer). The TE-aware overlay routing algorithm is given in Figure 2.3. The

first part, load-balancer, can be formalized as the following Linear Program model:

$$
\begin{aligned}
\min \quad & \sum_{s,t} d_{s,t}^{overlay} \\
\text{subject to} \quad & h_p^{s't'} \text{ is a logical routing} \\
& \forall \text{ link } l : t(l) = \sum_{s,t} f_{st}(l)(d_{st}^{under} + d_{st}^{overlay}) \\
& \forall \text{ link } l : t(l)^{overlay} = \sum_{s,t} d_{st}^{overlay} f_{st}(l) \\
& \forall s,t \in V : d_{s,t}^{overlay} = \sum_{s',t',p} \delta_p^{s't'} h_p^{(s't')} \\
& \sum_l \frac{t(l)^{overlay}}{cap(l) - t(l)} \leq \Theta
\end{aligned}
$$

Here, the main objective is to minimize the total overlay traffic amount. The Θ in the last constraint indicates the current latency.

Secondly, the limited selfish overlay routing can be defined as follows:

$$
\begin{aligned}
\min \quad & \sum_l \frac{t(l)^{overlay}}{cap(l) - t(l)} \\
\text{subject to} \quad & h_p^{s't'} \text{ is a logical routing} \\
& \forall \text{ link } l : t(l) = \sum_{s,t} f_{s,t}(l)(d_{st}^{under} + d_{st}^{overlay}) \\
& \forall \text{ link } l : t(l)^{overlay} = \sum_{s,t} d_{st}^{overlay} f_{st}(l) \\
& \forall s,t \in V : d_{st}^{overlay} = \sum_{s',t',p} \delta_p^{s't'} h_p^{(s't')} \\
& \forall \text{ link } l : \sum_{s,t} f_{st}(l) d_{st}^{overlay} \leq \theta
\end{aligned}
$$

Here, θ is the maximum link load that the overlay generates in the past run:

$$\theta = \max\{t(l)^{overlay} | \forall \text{ link } l\}.$$

2.6 Implementation

The vertical interaction game is implemented with a combination of General Algebraic Modeling System (GAMS) [3] programs and Perl scripts.

The General Algebraic Modeling System (GAMS) is specifically designed for modeling linear, nonlinear and mixed integer optimization problems. We use this modeling system to implement various optimization procedures for the experiments. The GAMS program is solved using COIN-OR solver (Computational Infrastructure - Operations Research), which is an initiative to spur the development of open-source software for the operations research community. The interaction between optimization programs is implemented by connecting the inputs and outputs of the GAMS programs through Perl scripts.

Given that we run the optimization process for more than hundred iterations, we need a support of Condor [2], which is a specialized workload management system for compute-intensive jobs. It provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management.

2.7 Simulation

This section describes the simulation results of vertical interactions. We first compare MPLS and COPE as the underlay traffic engineering schemes. Then we evaluate and compare TE-aware overlay routing and selfish overlay routing.

2.7.1 Data Set Description

We perform extensive experiments on a 14-node Tier-1 POP topology described in [28]. The underlay network topology is given in Figure 2.4. On top of the physical network, we made up a four-node full-meshed overlay networks as given in Figure 2.5. For the traffic matrix, we generate synthetic traffic demands using gravity model [28].

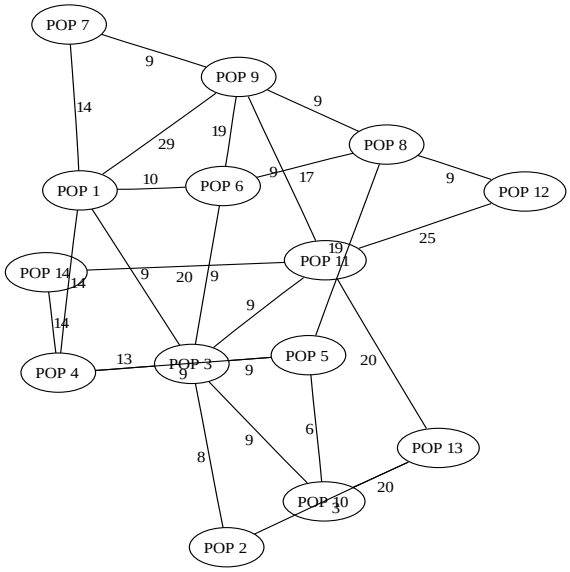

Figure 2.4: 14-node Tier-1 backbone topology: each node represents a Point-of-Presence (POP) and each link represents the aggregated connectivity between the routers belonging to a pair of adjacent POPs.

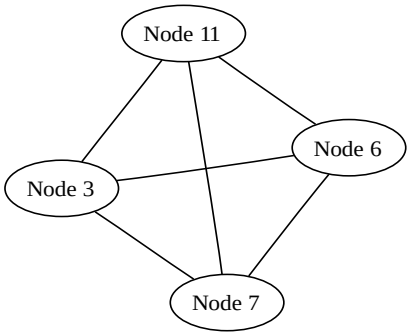

Figure 2.5: An overlay network used for the vertical interaction experiments. The overlay traffics are routed through the logical paths. In the simulation, we set all logical paths to include one transit point.

2.7.2 MPLS and COPE with Selfish Overlay

We start with the comparison between MPLS and COPE in the operator's viewpoint. We fix the overlay routing to be selfish and compare the performance of MPLS and COPE.

For the COPE, we need a prespecified penalty envelope value. We first calculate the value (1.9969) by running oblivious routing, which finds the optimal routing which minimizes the oblivious ratio. This can be calculated without any information about traffic demands because oblivious routing only depends on the network topology information. Then by multiplying 1.1 to the optimal oblivious ratio, we set the penalty envelope value.

First, we set 10% of the total traffic demand to be operated by the selfish overlay routing. We set the load scale factor to be 0.3, 0.5, and 0.7. This means that the maximum link utilization is 30%, 50%, and 70%, respectively, when all the demands use the default underlay routing without overlay's action.

The experiment results are shown in Figure 2.6. Regardless of the load scale factor, we can observe that the COPE makes better interaction with selfish overlay. In all cases, MPLS traffic engineering suffers from substantially large oscillation throughout the interaction, where COPE achieves almost stable performance with its maximum link utilization. Similarly, the dynamics of overlay latency is quite stable with the interaction of COPE. Moreover, the average latency sometimes gets improved by using COPE.

For the next experiment, we want to explore the impact of the overlay fraction to the vertical interaction. Now, we fix the load scale factor and change the fraction of overlay traffic (10%, 30%, 50%) in the experiment. We set the load scale factor to be 0.9 in Figure 2.7 and 1.0 in Figure 2.8.

Again, we can observe that COPE makes better interaction with selfish overlay routing than MPLS does. As we increase the fraction of overlay traffic, the oscillation of maximum link utilization gets larger, which follows our intuition. However, the performance of overlay latency seems to be independent of how much portion overlay routing operates.

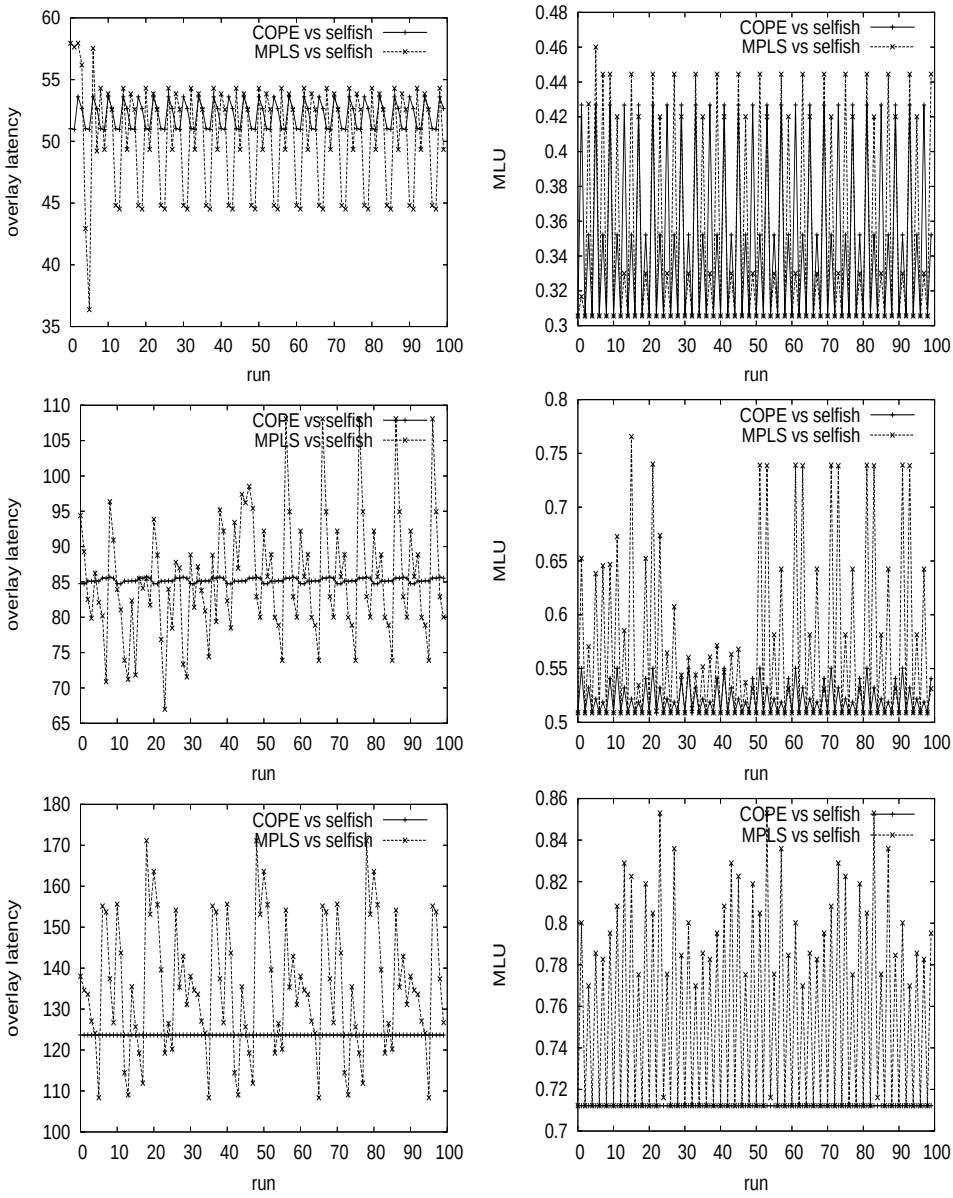

Figure 2.6: MPLS and COPE with selfish overlay. 14-node topology with a 4-node overlay network, overlay fraction = 10%, load scale factor = 0.3, 0.5, 0.7.

With the extensive simulation, we find COPE as a strong traffic engineering technique which achieves stable performance even the selfish overlay routing governs a significant portion of the total traffic demand.

2.7.3 TE-Aware Overlay and Selfish Overlay with MPLS

Now, we evaluate the TE-aware overlay routing by comparing it to the selfish overlay routing. For the underlay routing, we again use MPLS and COPE. We first start the evaluation by comparing two overlay routings on top of MPLS traffic engineering.

In Figure 2.9, we set 10% of the traffic to be operated by overlay routing and increase the load scale factor (0.3, 0.5, 0.7). Considering the overlay latency, TE-aware overlay routing achieves more stable performance. Moreover, in the case where the load scale factor is 0.5 and 0.7, the average latency experienced by TE-aware overlay is lower than selfish overlay. We can see that overlay routing can achieve better and stable routing by understanding the objective of underlay routing.

Considering the traffic engineering side, selfish overlay routing makes significant burden to the underlay routing because it generates substantially large amount of additional traffic. Thus, we can observe sudden increase of the maximum link utilization in all cases. However, TE-aware overlay limits its selfishness and tries to avoid a specific link to be over-loaded by its own traffic. Thus, the fluctuation of maximum link utilization is smaller when the overlay is TE-aware.

Next, we fix the load scale factor to be 0.9 and 1.0, and change the overlay fraction: 10%, 30%, and 50%. Figure 2.10 and 2.11 describe the results. The experiments are conducted where the network is substantially congested (load scale factor: 90% $\sim$ 120%). Still, the proposed method makes better interaction than selfish overlay does.

With the extensive experiment results, we come up with the conclusion that TE-aware overlay routing generally makes stable interaction with MPLS traffic engineering. Selfish overlay routing experiences less predictable latency and it makes significantly large

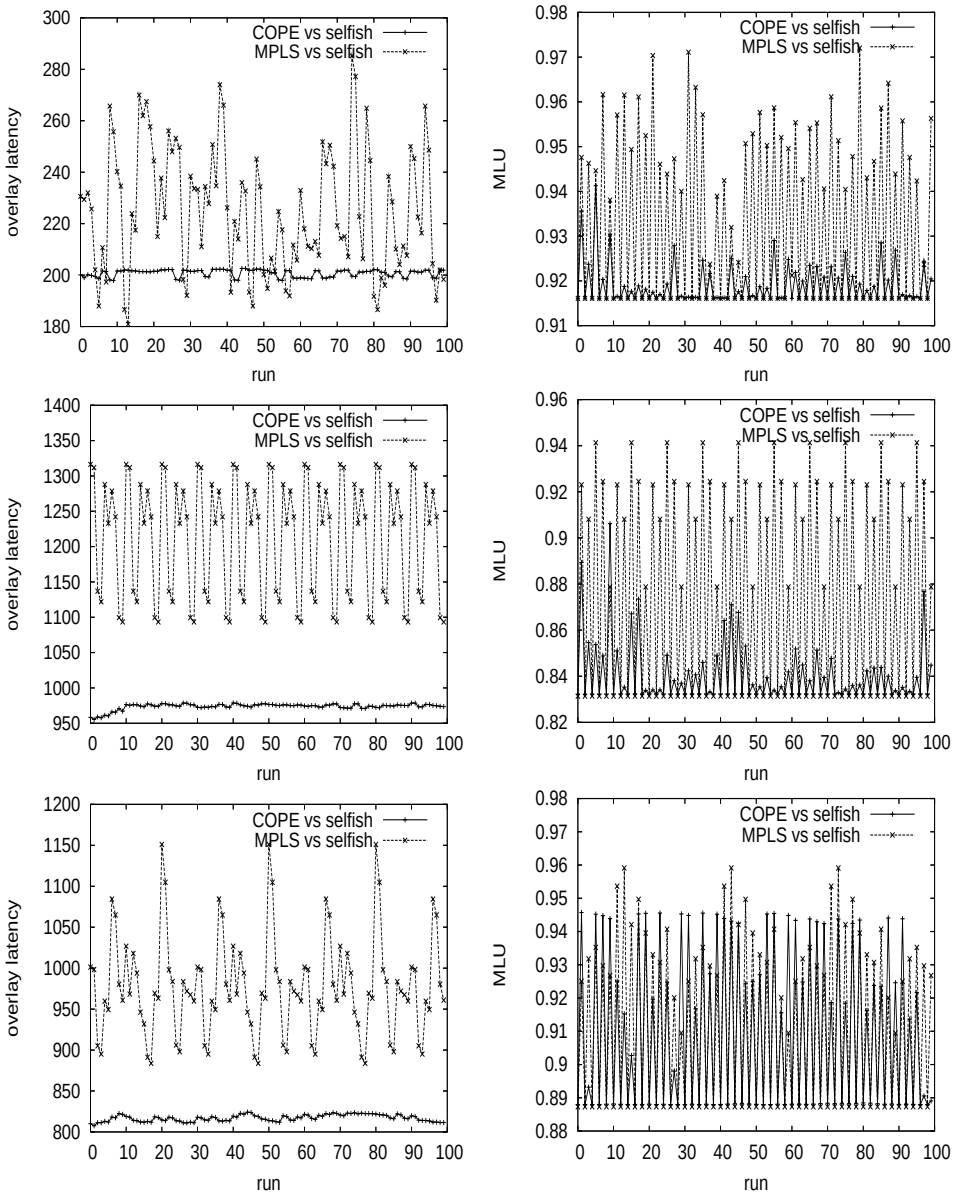

Figure 2.7: MPLS and COPE with selfish overlay. 14-node topology with a 4-node overlay network, overlay fraction = 10%, 30%, 50%, load scale factor = 0.9.

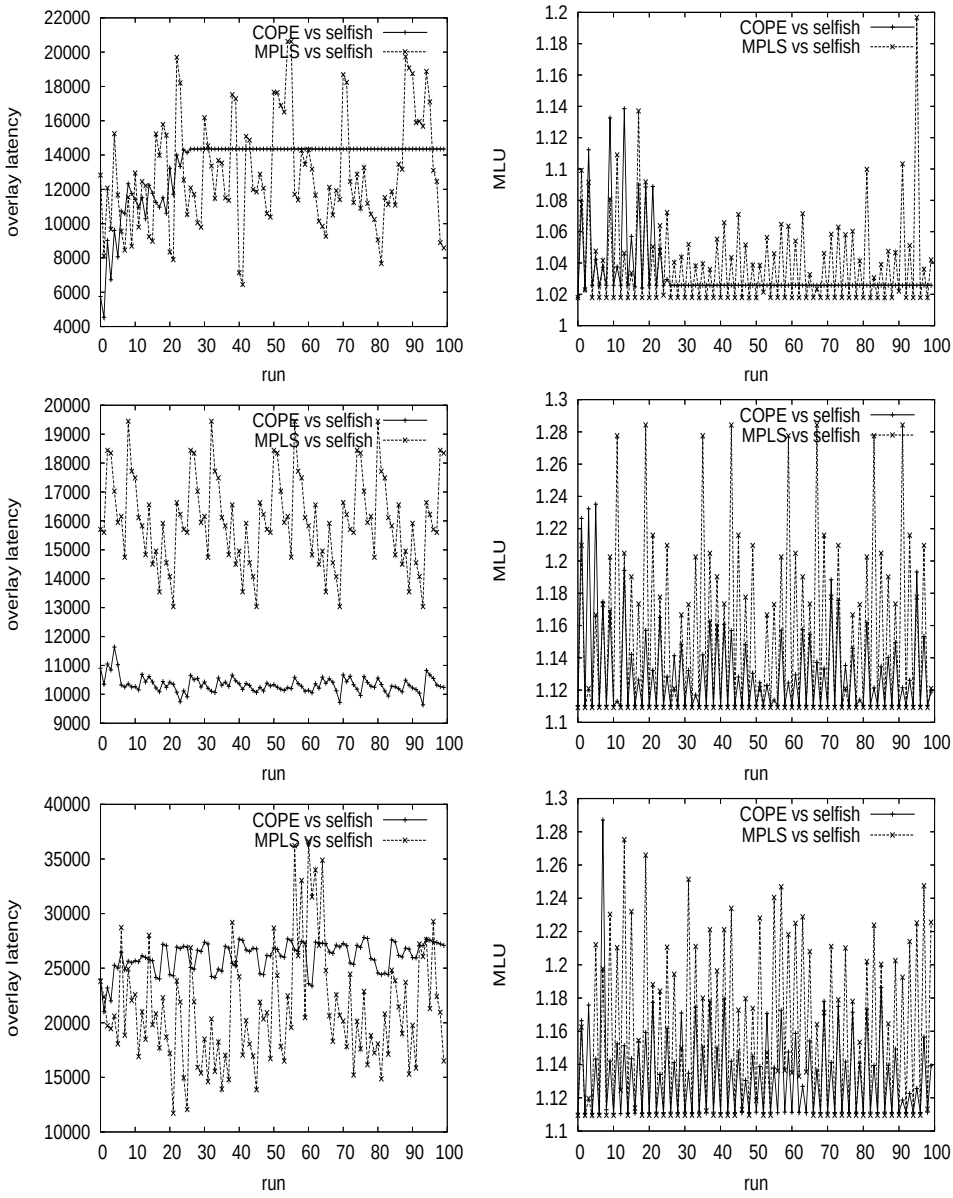

Figure 2.8: MPLS and COPE with selfish overlay. 14-node topology with a 4-node overlay network, overlay fraction = 10%, 30%, 50%, load scale factor = 1.0.

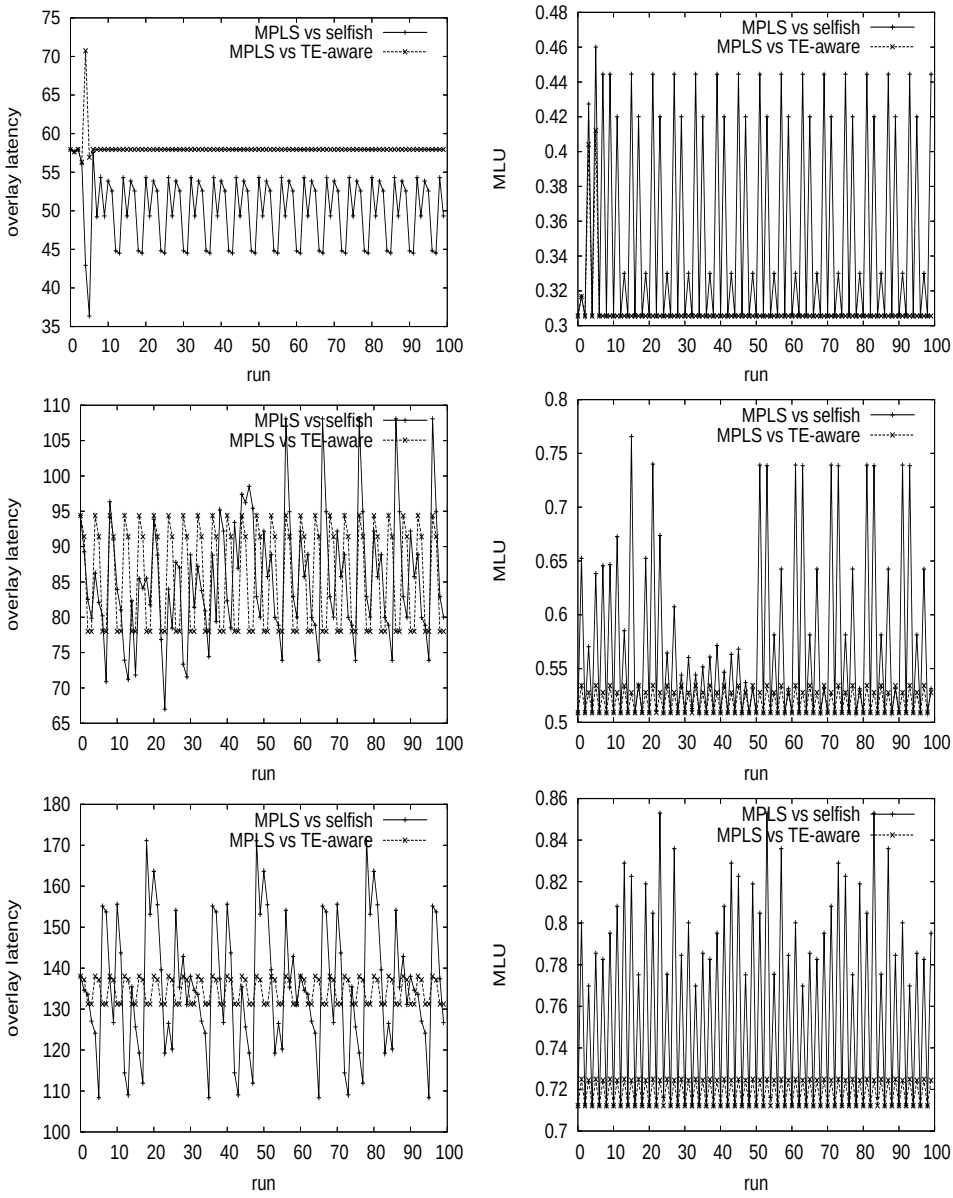

Figure 2.9: TE-aware overlay and selfish overlay on MPLS. 14-node topology with a 4-node overlay network, overlay fraction = 10%, load scale factor = 0.3, 0.5, 0.7.

maximum link utilization of the network. However, we can achieve either convergence or regular pattern with the overlay latency when TE-awareness is used in overlay routing. Moreover, the network overhead to the traffic engineering is reduced by using the proposed overlay routing. Thus, TE-awareness obtains win-win game for each player in the presence of MPLS traffic engineering.

2.7.4 TE-Aware Overlay and Selfish Overlay with COPE

For the last experiment, we examine the interaction between TE-aware overlay and selfish overlay on top of the COPE traffic engineering. In the previous experiments comparing COPE and MPLS, we have observed that COPE achieves better interaction with selfish overlay routing. Now, the question is how much gain we can get by using TE-aware overlay with COPE.

Figure 2.12 describes the experiment results, where 10% of the traffic is routed by overlay routing. We again use three load scale factors (0.3, 0.5, 0.7). Different from the experiments with MPLS, the achievement we get from TE-awareness is limited. When the load scale factor is 0.3 and 0.5, the TE-aware overlay routing converges fast with good latency, but TE-aware overlay routing shows a small oscillation in the last case. However, comparing to the oscillation in MPLS experiments, we can see the performance variation is negligible. Similar patterns can be observed with the maximum link utilization.

In Figure 2.13 and 2.14, we fix the load scale factors to be 0.9 and 1.0, and change the fraction of overlay traffic (10%, 30%, 50%). General observation is that as the link gets more utilized, the performance gain from TE-awareness is substantially large.

Considering the overlay side, in all experiments, the latency experienced by TE-aware overlay is better than that of selfish overlay. The maximum link latency experienced by selfish overlay is sometimes twice larger than average latency. In some scenario, the selfish overlay latency keeps increasing as the interaction with underlay proceeds. TE-aware overlay shows similar pattern but makes convergence at considerably lower latency. Look-

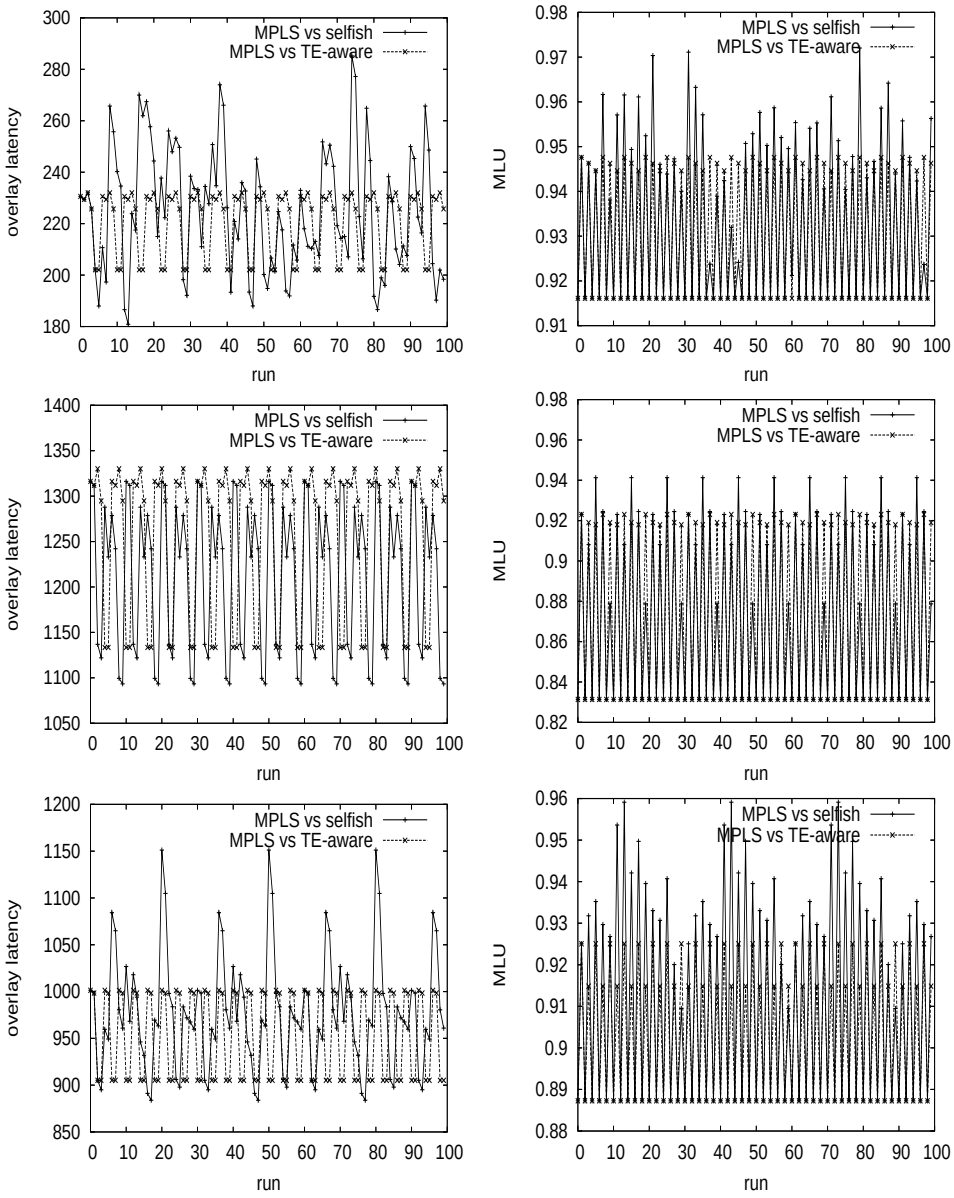

Figure 2.10: TE-aware overlay and selfish overlay on MPLS. 14-node topology with a 4-node overlay network, overlay fraction = 10%, 30%, 50%, load scale factor = 0.9.

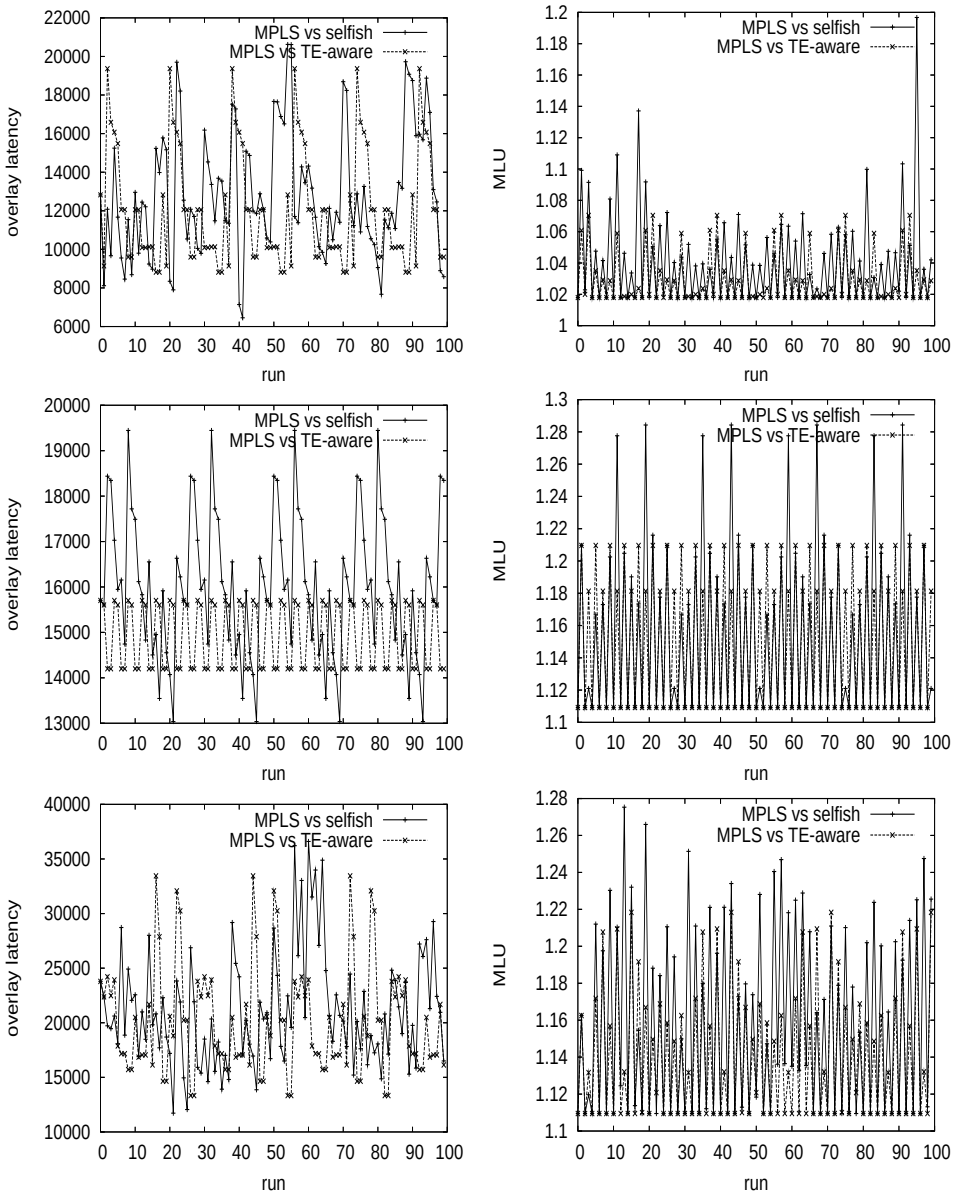

Figure 2.11: TE-aware overlay and selfish overlay on MPLS. 14-node topology with a 4-node overlay network, overlay fraction = 10%, 30%, 50%, load scale factor = 1.0.

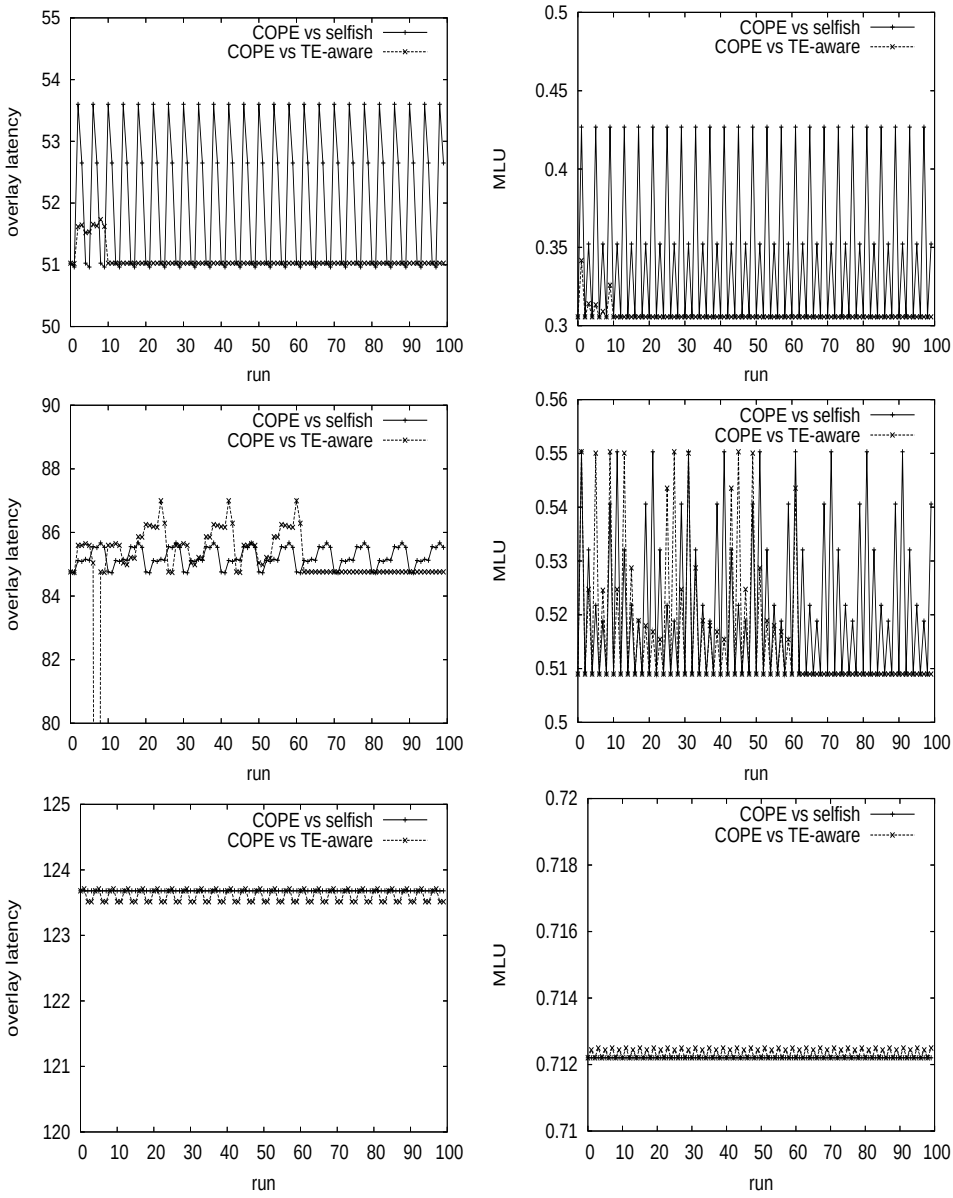

Figure 2.12: TE-aware overlay and selfish overlay on COPE. 14-node topology with a 4-node overlay network, overlay fraction = 10%, load scale factor = 0.3, 0.5, 0.7.

ing at the traffic engineering side, TE-awareness obtains either similar or better performance than selfishness.

Summarizing the interaction experiments with COPE, we can achieve considerably good interaction for both selfish overlay and TE-aware overlay. But, TE-aware overlay performs slightly better than selfish overlay routing.

2.8 Conclusion and Future Work

In this chapter, we study the interaction between overlay routing and traffic engineering. Overlay routing optimizes its logical routing with the given underlay routing, and this logical routing changes the physical demands observed by the underlay routing. Traffic engineering adapts its routing to cope with the new traffic matrix, which, in turn, changes the latency observed by overlay routing. We form this interaction as a non-cooperative two-player game, and observe that vertical interaction makes substantial oscillation, which degrades the performance for each player.

We implement and evaluate MPLS, oblivious routing, and COPE for the underlay routing. MPLS is an adaptive extreme in the sense that it tries to optimize the underlay routing with the current traffic matrix, and oblivious routing is the opposite extreme because it achieves some performance guarantee for all possible traffic matrices. We find COPE as an excellent combination of these two extreme methods. COPE achieves close-to-optimal performance for normal traffic demand, and, at the same time, it guarantees tolerable performance for extremely unpredictable traffic patterns. Our simulation shows that COPE performs well in the interaction with various overlay routing.

In addition, we propose *TE-awareness* to improve the current selfish overlay routing. The basic idea is to take traffic engineering's objective into account in overlay routing. If overlay is experiencing low latency, it tries to minimize the overlay traffic in the hope that this process will lighten the burden to traffic engineering. Otherwise, overlay tries to achieve better latency by changing the logical routing. Still, we make sure that overlay

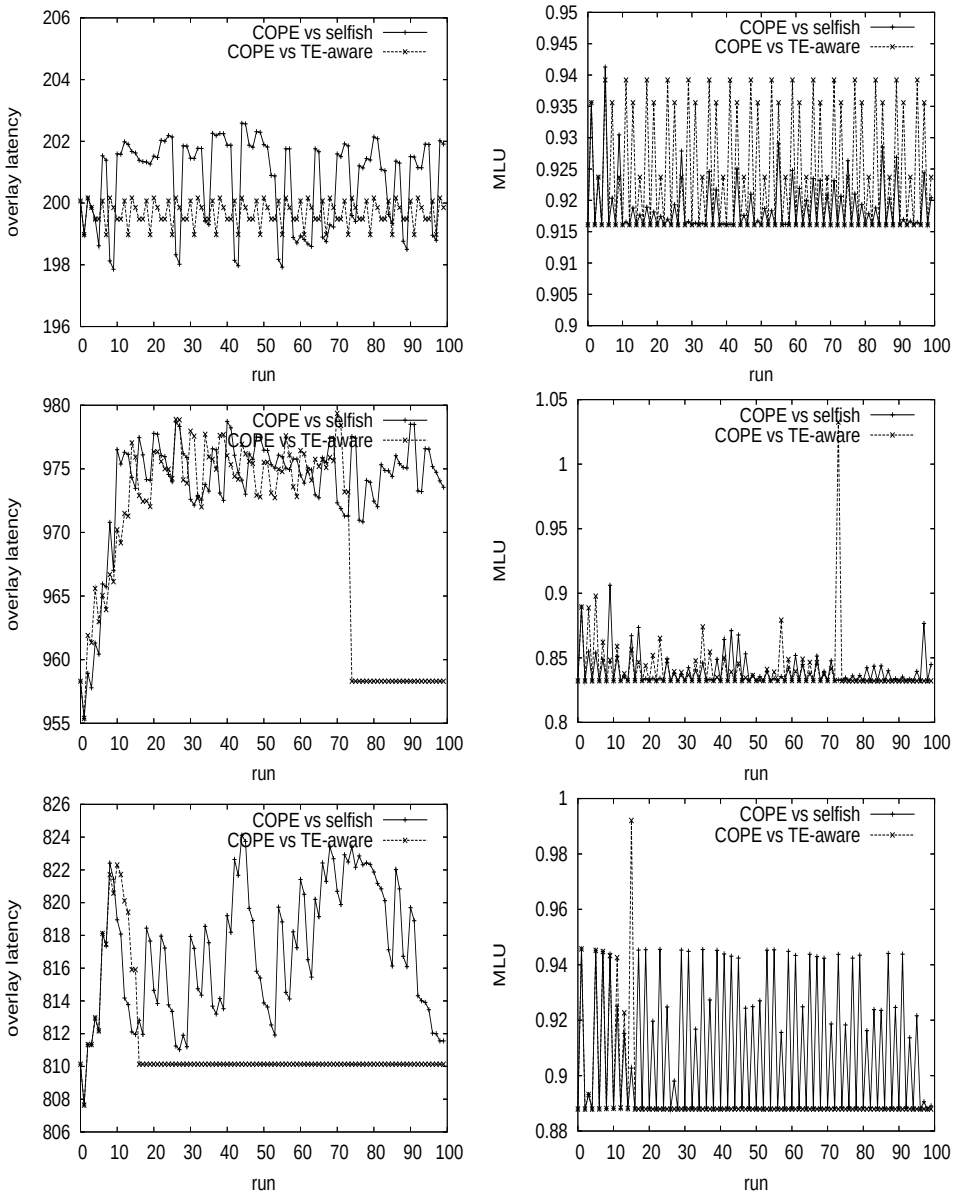

Figure 2.13: TE-aware overlay and selfish overlay on COPE. 14-node topology with a 4-node overlay network, overlay fraction = 10%, 30%, 50%, load scale factor = 0.9.

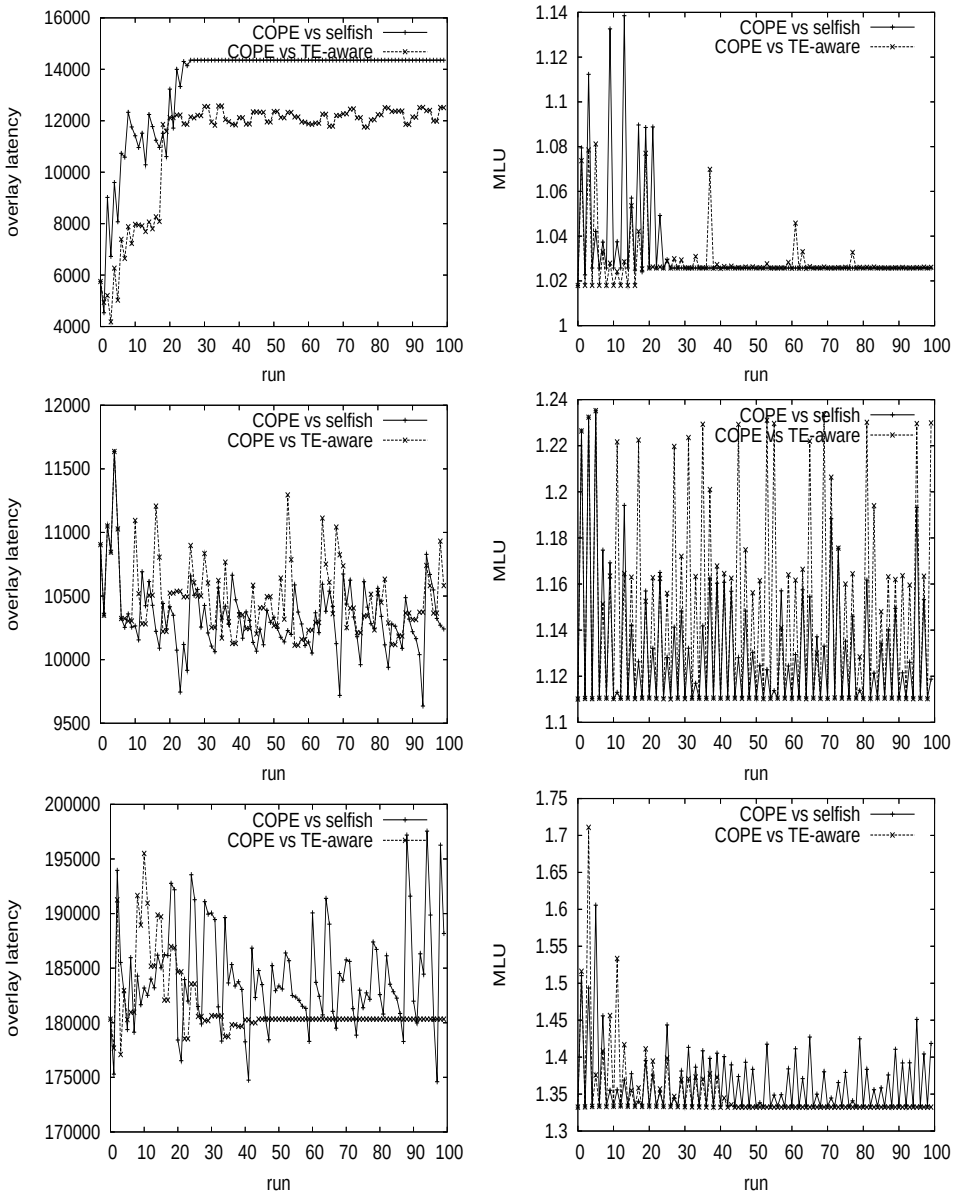

Figure 2.14: TE-aware overlay and selfish overlay on COPE. 14-node topology with a 4-node overlay network, overlay fraction = 10%, 30%, 50%, load scale factor = 1.0.

traffics do not overload a specific link so that overlay's behavior does not adversely affect the performance of traffic engineering.

With extensive simulation, we have shown that TE-awareness improves the vertical interaction. First, average latency experienced by TE-aware overlay is lower than that of selfish overlay. Moreover, the oscillation of latency gets improved significantly. In the underlay viewpoint, TE-awareness prevents the overlay to overload specific links. Thus, we can achieve win-win game with the proposed method.

In this chapter, we model the vertical interaction within a single domain. Then a future direction will be to explore the interaction in the inter-domain level. In this scenario, overlay networks spread across several autonomous systems and cooperate each other. Then the actions overlay take will affect multiple domains.

Chapter 3

Understanding the Interactions among Multiple Overlays

3.1 Introduction

Overlay-based approach has been successful because it can compensate the inefficiency of the current Internet without requiring modification in the IP infrastructure. Those overlay networks and peer-to-peer systems include Gnutella [4] (an overlay for file-sharing), Resilient Overlay Network [8] (which provides better connectivity than BGP does), and End System Multicast (which attempts to provide efficient multicast using end-hosts).

As overlay techniques get more popular, multiple overlay networks may simultaneously exist on a single physical network. Typically, overlay routing involves the measurement of all available overlay paths. However, if multiple overlays do not explicitly share their decisions on paths, the path information is likely to be inaccurate at the instant when the selection of the best route is made because of the actions by other overlays. Thus, the decisions made by one overlay network can directly affect the performance of other overlays. We term this as *horizontal interaction* of overlay routing.

There have been several research works on the dynamics of multiple overlays. Qiu

et al. [32] first address interactions between multiple coexisting overlays, where the authors investigate the performance of selfish routing after the system reaches the Nash equilibrium point. Seshadri and Katz [35] investigate the performance of greedy route selection in a variety of scenarios. Their finding is that greedy overlay routing can perform poorly if the overlay flows comprise a significant fraction of link capacities. Jiang et al. [22] model the interaction of overlays as a non-cooperative strategic game and prove the existence of a Nash equilibrium in the game. It is shown that the outcome of the game may not be Pareto-optimal in some scenarios. Keralapura et al. [23] describe how multiple similar or dissimilar overlay networks could experience race conditions, resulting oscillations in route selection and network load.

In this chapter, we first start with the *pure* horizontal interaction, where two overlays change their logical routing on top of a *fixed* underlay routing. In this scenario, we assume that individual overlays do not recognize the existence of the other parties. We find that, when the links are under-utilized with small traffic demands, the interaction performs efficiently without significant oscillation issues. However, as the total traffic is increased and network gets congested, the interaction makes oscillation for both overlays. Next, we seek to understand the interaction of multiple overlays on top of dynamic underlay routing. Three players are involved in this game: two overlay optimizers and an underlay TE optimizer. The simulation shows that the severe oscillation degrades the performance of all players. However, we find that TE-awareness and demand-obliviousness can improve the interaction.

The chapter is organized as follows. We first formally define the horizontal interaction as a non-cooperative game in Section 3.2. Then we show the preliminary simulation results in Section 3.3 and conclude the chapter in Section 3.4.

3.2 Horizontal Interaction Game

This section formulates the horizontal interaction as a non-cooperative game among overlay networks. The underlying mathematical model is essentially identical to that of vertical interaction. Refer to Table 2.1 for the details. The only difference is that we need indexed notations for each overlay network, rather than having a single group of notations.

3.2.1 Pure Horizontal Interaction

Now, we assume two overlays exist on top of a shared underlay topology, and that overlays do not have information about each other. In the *pure* horizontal interaction, the underlay routing is fixed and only overlays change their logical routing. Each overlay makes *simultaneous* move based on the given network condition, assuming that all the other factors related to network are static. Note that, in the vertical interaction game, overlay routing and traffic engineering make *sequential* moves based on the other party's action.

Given the physical network topology, underlay routing, and experienced latency for each link, optimal logical routing of overlay 1, $\{h_p^{s't'} | \forall s', t' \in V', \forall p \in P^{(s't')}\}$, can be obtained by solving the following non-linear optimization problem:

$$
\begin{aligned}
\min \quad & \sum_l \frac{t(l)^{overlay_1}}{cap(l) - t(l)} \\
\text{subject to} \quad & h_p^{s't'} \text{ is a logical routing} \\
& \forall \text{ link } l : t(l) = \sum_{s,t} f_{st}(l)(d_{st}^{under'} + d_{st}^{overlay_1}) \\
& \forall \text{ link } l : t(l)^{overlay_1} = \sum_{s,t} d_{st}^{overlay_1} f_{st}(l) \\
& \forall s, t \in V : d_{st}^{overlay_1} = \sum_{s',t',p} \delta_p^{s't'} h_p^{(s't')} \\
& \forall s, t \in V : d_{st}^{under'} = d_{st}^{under} + d_{st}^{overlay_2}
\end{aligned}
$$

The first constraint ensures that the routing satisfies the flow conservation law. The last

constraint is needed because overlay 1 cannot differentiate the traffic of overlay 2 from the original underlay traffic. Specifically, d_{st}^{under} indicates the original underlay traffic, and d_{st}^{under} and $d_{st}^{overlay_2}$ add up to make $d_{st}^{under'}$, the cross traffic observed by overlay 1. Then, the non-linear objective function is again linearized with the same arguments in Chapter 2. For overlay 2, we can easily have similar optimization program.

3.2.2 Combining Horizontal and Vertical Interaction

We now add the vertical interaction process to the pure horizontal interaction game. That is, traffic engineering adaptively changes the physical routing based on the current traffic matrix. MPLS [33] and COPE [39] are used for the traffic engineering algorithm. When the traffic engineering modifies the underlay routing, the latency observed by the overlays are also changed. Then overlays simultaneously optimize their routing, which, in turn, change the traffic matrix observed by network operating side.

3.3 Simulation

This section describes the simulation results of horizontal interactions among overlays.

3.3.1 Implementation

Similar to the vertical interaction experiments, we implement various optimization problems in GAMS [3] programs, then we develop Perl scripts to make the interaction between GAMS programs. Condor [2] system is used to execute the intensive jobs.

3.3.2 Data Set Description

In the simulation, we assume that two overlay networks coexist on top of the shared underlay network. The logical topology for each overlay is given in Figure 3.1. Two overlay nodes are intentionally set to be included in both overlays, and they share a link between

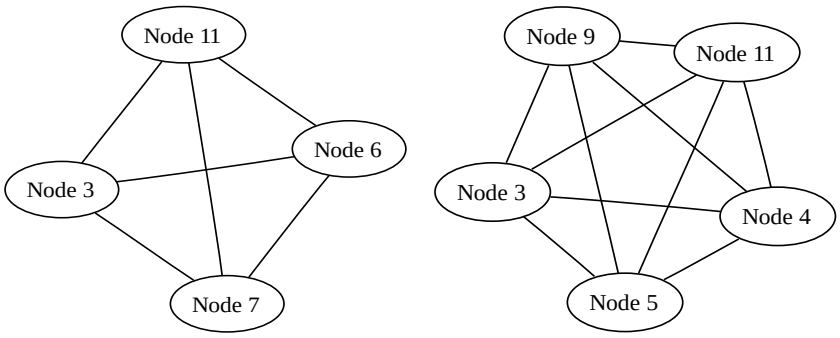

Figure 3.1: Two overlay networks that are used for the horizontal interaction experiments. Both overlays do not know the presence of each other in this simulation.

node 3 and 11. The first overlay has four nodes and the second has five. For the physical network topology, we again use the 14-node Tier-1 POP topology described in [28]. For the traffic matrix, synthetic data based on gravity model [28] is generated. In order to ensure the fair condition for each overlay network, two overlays are set to have similar total traffic demands.

3.3.3 Horizontal Interaction on Static Underlay Routing

First, we start with pure horizontal interaction where only overlay networks change their routing with static underlay routing. We set both overlays to use selfish optimizers.

In Figure 3.2, we use MPLS traffic engineering to calculate the underlay routing. In the simulation, the overlay demands totally account for 20% of the total traffic demands, and three experiments are done with different load scale factors (0.4, 0.7, 1.0). When the network is under-utilized, the interaction converges really fast without oscillation issues. However, in the case the maximum link utilization is over 100%, the horizontal interaction results some oscillation. Similar results are observed in Figure 3.3, where the underlay routing is given by COPE.

For the next experiment, we evaluate the TE-aware overlay routing in the pure horizontal interaction. The preliminary results indicate that TE-awareness makes little impact

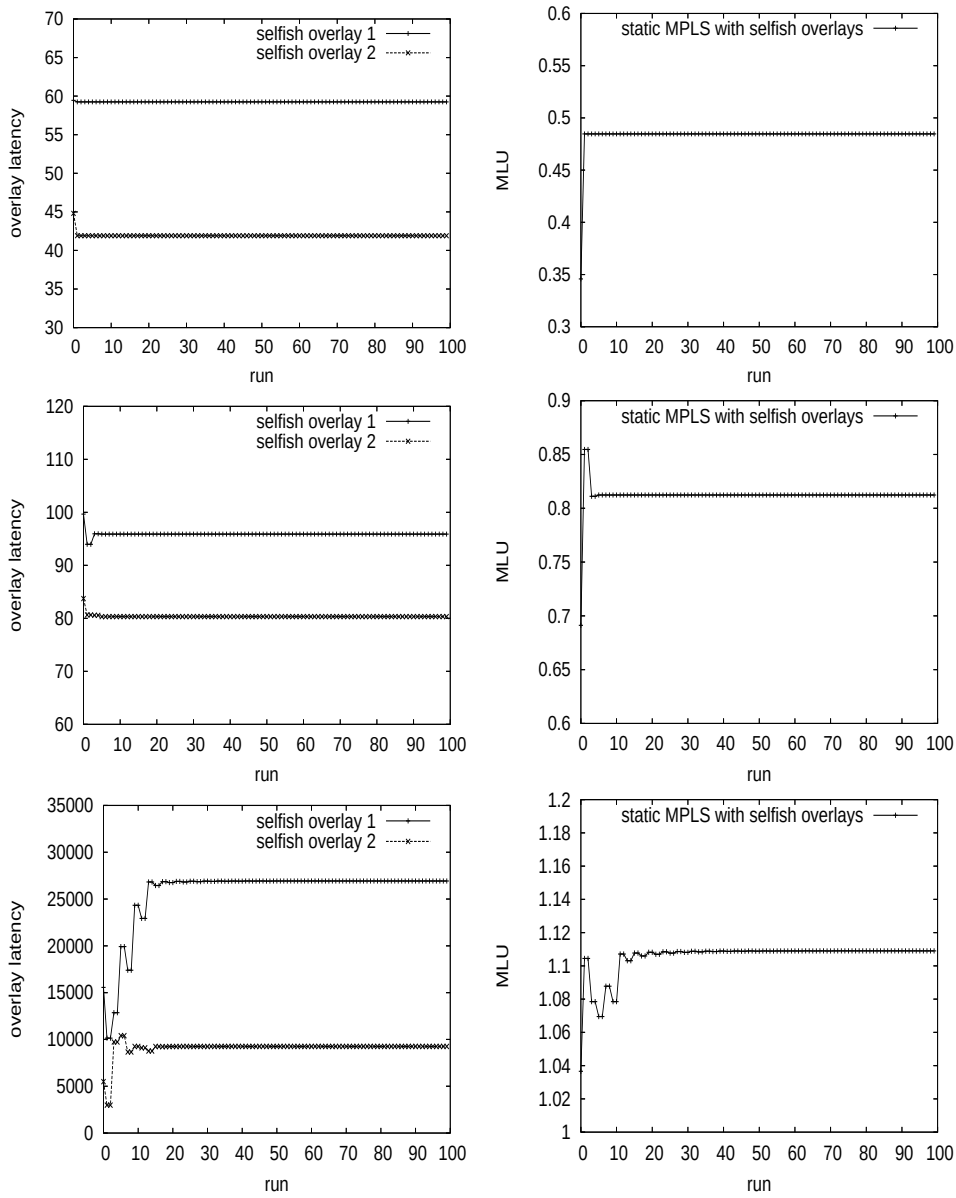

Figure 3.2: Horizontal interaction of two selfish overlays on static MPLS. Total overlay fraction = 20%, load scale factor = 0.4,0.7,1.0.

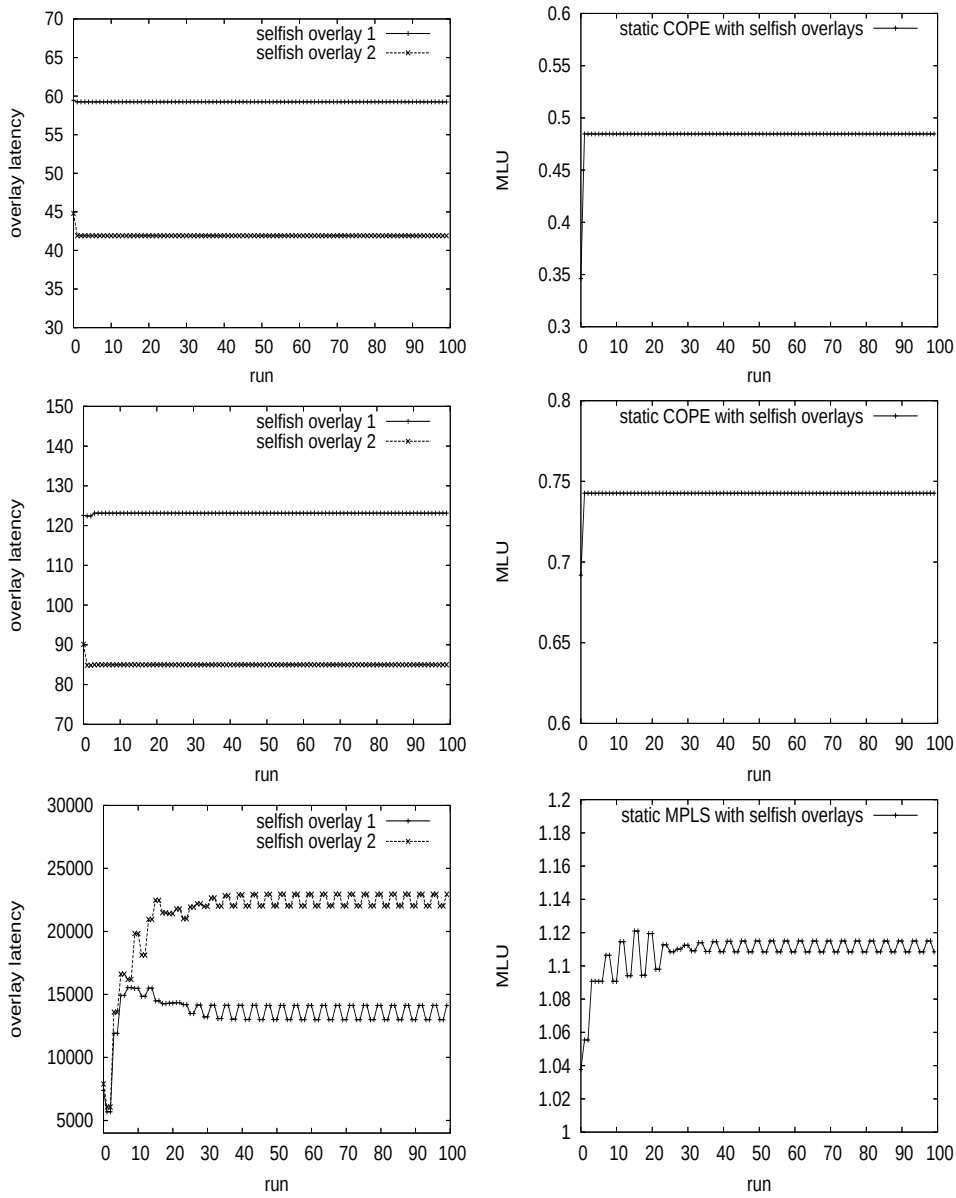

Figure 3.3: Horizontal interaction of two selfish overlays on static COPE. Total overlay fraction = 20%, load scale factor = 0.4,0.7,1.0.

on the game. When TE-aware overlays are used on top of the static underlay routing, the qualitative results are similar to that of the experiments with selfish overlays. Thus, we do not include those results.

In the pure interaction game, our general finding is that horizontal interaction itself does not make severe oscillation problem, comparing to the vertical interaction. To support our argument, we will further explore the interaction in other scenarios with different topologies.

3.3.4 Horizontal Interaction on Adaptive Underlay Routing

Now, we combine horizontal interaction and vertical interaction. In this scenario, traffic engineering adaptively change its routing to cope with the new traffic matrix, then two overlays make simultaneous moves to adapt their logical routing to the new physical routing.

In Figure 3.4, we fix the underlay routing to use MPLS traffic engineering and compare the performance of TE-aware overlays against that of selfish overlays. We use two different settings where the load scale factors are 0.5 and 0.7, and 20% of the total traffic is set to be governed by overlay routing in each case. Consistent observation in all results is that selfish overlays make bigger, sudden spikes throughout the interaction and that TE-aware overlays achieve more stable latency performance with faster convergence. Moreover, selfish overlays significantly increase the total traffic amount and the maximum link utilization sometimes increases by a factor of two, which makes negative impact to the network operating side.

Figure 3.5 describes the experiment results where the underlay routing is determined by COPE optimizer. All the experiment parameters are identical to that of the previous experiment in Figure 3.4 (load scale factor = 0.5, 0.7 with 20% overlay traffic demand). Consistent with the results in vertical interaction experiments, COPE generally achieves much better interaction with both selfish and TE-aware overlays. Given that the interaction

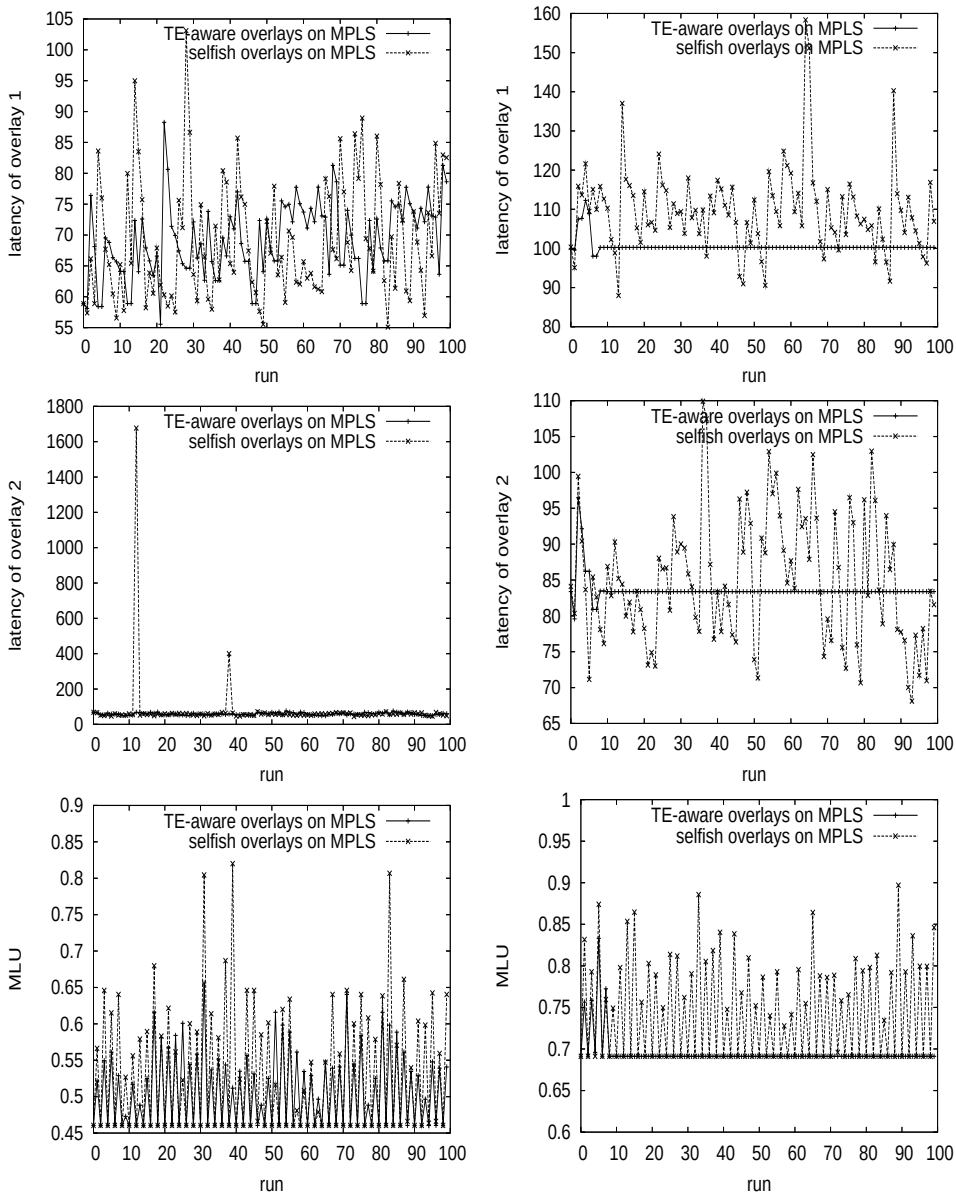

Figure 3.4: TE-aware overlays vs selfish overlays on adaptive MPLS. Total overlay fraction = 20%, load scale factor = 0.5, 0.7.

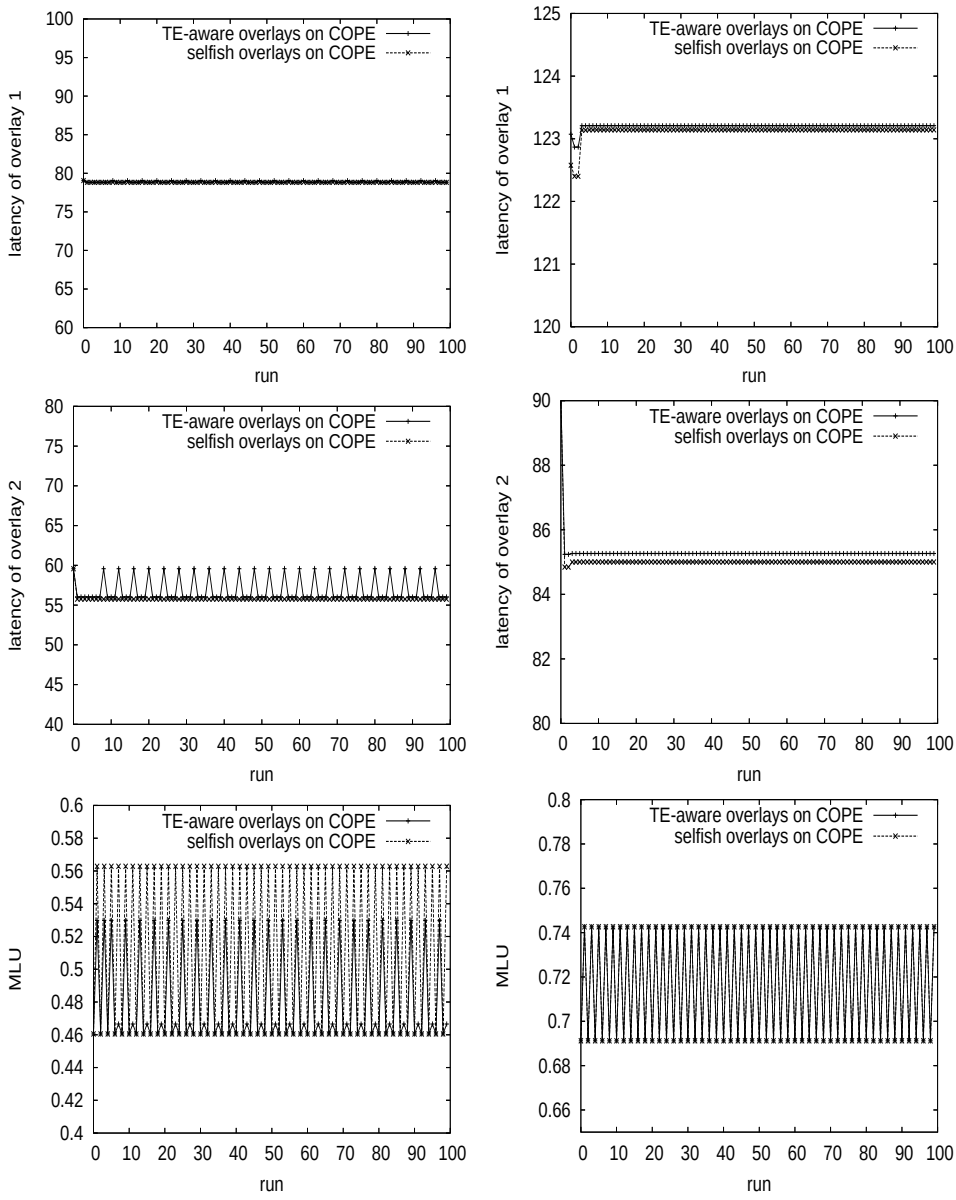

Figure 3.5: TE-aware overlays vs selfish overlays on adaptive COPE. Total overlay fraction = 20%, load scale factor = 0.5, 0.7.

is already improved, TE-awareness does not significantly change the quality of interaction.

So far, we assume that both overlays are either selfish or TE-aware. We have also simulated the case where one of overlays is selfish and the other is TE-aware. Our preliminary results indicate that the improvement achieved by TE-awareness in the case is quite limited. A potential explanation is that selfish overlay overloads some specific links and these congested links severely degrade the performance of both overlay networks. We argue that each overlay network has an incentive to be TE-aware because selfishness will hurt its own performance, as well as others. We will further explore the incentives of overlays to be TE-aware with extensive experiments.

3.4 Conclusion and Future Work

This chapter gives initial results on the horizontal interaction of overlay routing. As overlay network gets more popularity, multiple overlay networks may coexist on top of a shared underlay routing without knowing the presence of each other. Each overlay network periodically probes the available overlay paths and changes its logical routing to reduce the total latency. However, if multiple overlays share some logical links without exchanging explicit routing information of each other, currently observed link latency may be changed by other overlays' action. Thus, the action of each overlay affects other overlays. We form this interaction as a non-cooperative game among multiple overlays.

The simulation results show that, if the underlay routing is static, horizontal interaction itself does not make severe oscillation problems. Even in the case where the network is sufficiently congested, the oscillation does not last for a long period.

We then include the vertical interaction process into the horizontal interaction. Multiple overlays optimize their routing on top of an adaptive underlay routing. COPE makes better interaction with any combinations of overlay routing optimizers than MPLS does, which is consistent with the results in Chapter 2. In addition, TE-awareness improves the interaction process when all overlay networks choose to be TE-aware. However, when one

of the overlays decides to be selfish, the overall performance gain with TE-awareness seems to be limited, and the selfish overlay itself also suffers from the poor interaction in this case. We argue that every overlay network has an incentive to be TE-aware because it can achieve better interaction by limiting its selfishness.

This chapter focuses on a single-AS scenario, and a natural future direction is to extend the framework to inter-domain level. We can obtain more realistic experiments if overlay nodes are located in multiple domains. Another direction is to incorporate network anomalies in the scenario where some links fail in the middle of interaction process. It will be interesting to observe how the network performs in the transient phase and how fast the interaction converges to the stable state.

Chapter 4

Designing an Incentive-based Framework for Overlay Routing

4.1 Introduction

Overlay networks have been successful for many applications such as content distribution networks, peer-to-peer file sharing, ad-hoc networks, distributed look-up services, application-layer multicast overlays, virtual private networks, and so on. Especially, it has been shown by [8, 34] that overlay routing can significantly improve the sub-optimality of the Internet routing. Most overlay systems assume that individual nodes will cooperate with each other to achieve the goal of the whole system.

However, as the overlay network becomes more popular, we argue that the system will get decentralized and it will consist of independent overlay nodes. The node will choose the action with its own decision. We assume that an overlay node will selectively forward traffic from other nodes to maximize its benefit and to minimize the cost. The benefit a node can achieve is better routes with lower latency and lower loss rate. In the meantime, the node will consume some resource when forwarding transit traffic on behalf of other nodes. The cost includes packet processing time, memory consumption, and network

bandwidth.

We analyze the behavior of the overlay nodes by formalizing the transit traffic forwarding as a non-cooperative game [17]. Since the nodes in overlay routing are network routers, we assume that the identities of the nodes are well-known to each other and that every pairs of nodes will repeat the game. Our analysis shows that players are not likely to cooperate if there is no regulation mechanism.

In this chapter, we introduce incentive-based frameworks [17], which are designed to stimulate the players to cooperate in overlay routing. We analyze three trigger strategies for the repeated game: grim-trigger, tit-for-tat, and punish-and-reward. We give sufficient conditions, in which the independent nodes follow the rule of the systems. In addition, we generalize the punish-and-reward system to induce more cooperation of the players.

Among the possible strategies, we evaluate the generalized punish-and-reward system with simulation. The system stimulates significant amount of cooperations with limited number of punishments. The performance gets closer to the social optimum as we have more cooperation. In addition, the framework shows good tolerance against impatient players.

To the best of our knowledge, there is no general framework to stimulate cooperations in overlay routing networks. Considering an increasing role of overlay networks, proposed mechanism will merit general purpose systems of overlay networks.

The chapter is organized as follows. First, we assume our model and define the problem in Section 4.2. Then we use game-theoretic approach to analyze the behavior of the overlay nodes in Section 4.3. Section 4.4 proposes our framework to make incentives to cooperate in the overlay routing, and Section 4.5 evaluates the proposed method using simulation. We discuss related research works in Section 4.6, and conclude the chapter with future direction in Section 4.7.

4.2 Our Incentive Model

In this section, we introduce requirements and assumptions which the model is based on. Then the transit traffic forwarding problem in overlay routing is formalized in a game-theoretic framework.

4.2.1 Requirements and Assumptions

Our incentive model is based on the assumptions and requirements as follows:

1. Quantifying benefits and costs: Overlay nodes can get better routing paths with the help of other nodes. The benefit we can achieve includes lower latency and lower loss rate. We may precompute these benefit values using network coordinate systems [16]. We assume that all the players know the benefit for each opponent. In the meantime, overlay nodes spend some resource to forward traffic from their opponents. The cost includes CPU time, memory consumption, and network bandwidth. Based on the current demand, each overlay node calculates the cost that will occur to the opponent and announces this information to the opponent. Since we are dealing with different kinds of metric, we need to convert them into dimensionless parameters.

2. Strategic and rational overlay nodes: The participants are strategic in the sense that they can choose their own actions whatever they want, and that these actions lead to different outcomes. A node can take action to forward or to drop the traffic from a given opponent node. In addition, overlay nodes are rational to maximize their own utility values.

3. Overlay nodes are trustworthy: We assume that each overlay node will truthfully report the cost that will occur to forward the transit traffic and the benefit the node can get from the opponent. It will be an interesting problem to think about the incentive of players to truthfully report them. However, it is not a focus of this chapter. Another assumption is that if an overlay node agrees to relay other's traffic, it will actually put its best effort to deliver the traffic to the destination. Since overlay nodes in our context are routers, we exclude the case where some players are malicious.

		Player 2	
		Relay	Not Relay
Player 1	Relay	$b_{1,2}$-$c_{1,2}$/$b_{2,1}$-$c_{2,1}$	-$c_{1,2}$/$b_{2,1}$
	Not Relay	$b_{1,2}$/-$c_{2,1}$	0/0

Figure 4.1: Traffic relaying game in overlay routing is equivalent to Prisoner's Dilemma

4. Identities of nodes: The overlay nodes are routers and the identities are well known to each other. Consequently, the relationship among participants is continuing. However, in P2P system, the up-time of the clients is relatively short, which makes it difficult to regulate anonymous free riders. In this sense, P2P system can be modeled as a one-shot game, but repeated game framework should be applied to the overlay routing.

4.2.2 Problem Definition

In this chapter, we want to answer the following questions. *What will be the overall performance if overlay nodes are independent?* If the overlay nodes can be controlled by a central entity, we may achieve the social optimal performance. However, as the overlay nodes seek individual optimality, it is likely that the overall performance degrades (price of anarchy). In Section 4.3, we will analyze the transit traffic forwarding as a non-cooperative game. Given the overlay network experiences performance degrading by the selfishness, then the next question is: *how can we design a mechanism to encourage overlay nodes to cooperate each other and to get closer to socially optimal performance?* In Section 4.4, we propose our framework to solve this problem.

4.3 Traffic Relaying Game

In this section, we want to anticipate what will happen if the overlay nodes are independent players. Since the individual action of each player will lead to the final outcome, we use

N	total number of overlay nodes
i, j	individual overlay node
u_{ij}	utility value node i achieves from j's contribution
c_{ij}	cost of node i to relay j's traffic
b_{ij}	benefit node i gets from j's help
δ	discount factor

Table 4.1: Notations for internal interaction

game theory to analyze the overlay routing.

Since there is no standard consensus on the overlay network construction process, we describe the procedure we will use in this chapter. (1) When a router joins the overlay, it receives the protocol of the overlay network. (2) With the current traffic demand, each node calculates the benefit it can achieve from other nodes. (3) Each node sends traffic relaying requests to the opponents with the cost information including network bandwidth and traffic amount. (4) Based on the benefit and cost values, each pairs of nodes play traffic forwarding game.

Figure 4.1 shows the transit traffic forwarding game in the overlay routing. We assume that both players simultaneously select their actions: to relay the opponent's traffic or not. If both players forward the traffic on behalf of each other, both get the benefit from each other and consume their resource for each other. If only one of the players relays the traffic, the cooperative player pays the cost without benefit and the other player receives the benefit without any cost.

Here we introduce some notations for our convenience (Table 4.1). We assume that there are N players and $\frac{N(N-1)}{2}$ separate games among them. In the game between player i and player j, we denote the partial utility function of player i to be $u_{ij} = -c_{ij} + b_{ij}$, where c_{ij} is the cost player i should pay to relay player j's traffic and b_{ij} is the benefit player i can get from player j's help. Then total utility function of player i is $\sum_{i \neq j} u_{ij}$. We may use b and c without the subscripts if there is no confusion.

Let's consider the best action each player can take in this game. Basically, this game

is equivalent to Prinsoner's Dilemma [17]. Assuming player 2 relays the traffic, player 1 gets better payoff by not relaying the traffic ($b_{12} > b_{12} - c_{12}$). With the assumption that player 2 does not help out, player 1 is still better off to refuse the duty ($0 > -c_{12}$). Similarly, player 2 will not cooperate with the same argument. Thus, $(NotRelay, NotRelay)$ is a *Nash equilibrium* [17, 29] of this game. The players are said to be at Nash equilibrium if no player can improve its utility by unilaterally changing his strategy.

With this argument, we can see that overlay nodes are unlikely to cooperate without some external control. Thus, we need some mechanism to give incentives to players to cooperate. Intuitive approach could be to record the history of other players' actions and maintain the reputation value of the opponent player.

We model the traffic forwarding in overlay routing as a repeated game because the overlay nodes are network routers and the interaction between nodes will be repeated. Since we do not expect the end of the interaction, we treat it as an infinitely-repeated game. This feature is different from the case of P2P file-sharing system, where anonymous players have incentives to get the download and to leave the system without any contribution.

In the repeated game, overlay nodes play the game in Figure 4.1 over and over. The utility value is simply the total sum of the utility the player gets in a one-shot game. Since the game is played infinitely, we have to differentiate between the utility achieved in the current time and the one in the future games. *Discount factor* ($0 \leq \delta \leq 1$) is the weight we put on the future utility. If $\delta = 0$, the player only considers the current payoff and ignores the future impact. If δ gets closer to 1, the player puts almost equal importance on the future payoff as the present value.

4.4 Incentive-Based Frameworks

In this section, we introduce possible frameworks adopted from game theory [17]. They are designed to encourage cooperation in overlay routing. We itemize the requirements of the framework we want to achieve. Then we describe some candidate systems and evaluate

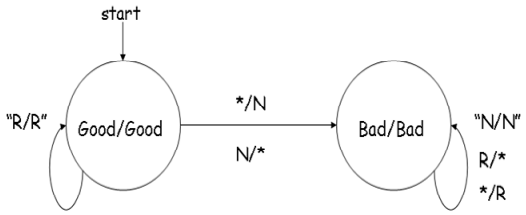

Figure 4.2: Grim-Trigger State Diagram

them using mathematical arguments.

The followings are some requirements that the system should meet. (1) The system stimulates the independent players to cooperate each other. (2) The system is robust with the "impatient" players (i.e. lower δ value). (3) The outcome of the system is close to the social optimal performance. We have three possible systems for our goal, and details will be cover in the following sections.

- Grim-Trigger: If both players cooperate in the past, they repeat the cooperation. However, if there is a player who defected, there will be no cooperation thereafter.
- Tit-for-Tat: In this system, each player repeats whatever the opponent played in the previous game. This follows the human analogy "an eye for an eye, a tooth for a tooth".
- Punish-and-Reward: We consider the notion of "forgiveness" in this system, whereas grim-trigger only includes punishing mechanism. A player in a bad reputation can restore its state by receiving the punishment.

4.4.1 Grim-Trigger System

In grim-trigger system, everyone enjoys the cooperation until there is one player who defects at some point. Figure 4.2 shows the state diagram of the system. The quoted actions are what our rule requires the players to obey. Both players start with the good state as long

as both of them relay the opponent's traffic. But if one of them refused to relay at some point, both of them go to bad state and stay there forever.

The grim-trigger is strong and extreme because there is no forgiveness. If one of the players refuses to cooperate, both players will not cooperate forever. Therefore, players will be really cautious to refuse the cooperation.

Since the overlay nodes are independent, they can choose whether to go along with the system or not, based on the expected payoffs. The following proposition gives a condition where players will follow the rule.

Proposition 1 *Grim-Trigger System induces the cooperation of the players as a subgame perfect Nash equilibrium if* $\delta \geq c/b$.

Proof: It is shown in [17] that it is enough to consider one time unilateral deviation to show it as a Nash equilibrium. For each state, we consider the utility payoff each player can have by obeying or deviating, assuming that both players will obey the rule thereafter.

Suppose players are in the good state. Then the rule is to play (R, R). *If player 1 obeys the system* (R, R), *both players will stay in the good state. If player 1 deviates from the rule* (N, R), *there will be no cooperation forever.*

$$\begin{aligned} u_1(obey|GG) &= \sum_{i=0}^{\infty} (b-c)\delta^i = \frac{b-c}{1-\delta} \\ u_1(deviate|GG) &= b + \sum_{i=1}^{\infty} 0\delta^i = b \end{aligned}$$

Player 1 will obey the rule if $\delta \geq c/b$ *in the good state. We can have a dual argument for player 2.*

Suppose players are in the bad state. Then the rule is to play (N, N). *Regardless*

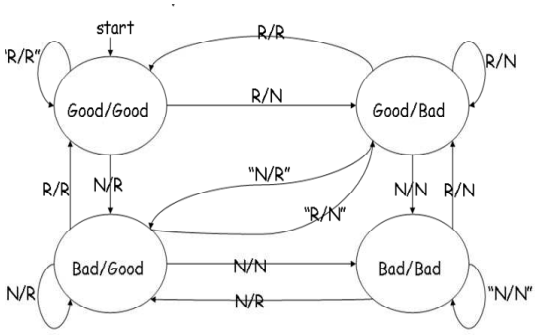

Figure 4.3: Tit-for-Tat State Diagram

of player 1's actions, they will stay there forever.

$$u_1(obey|BB) = \sum_{i=0}^{\infty} 0\delta^i = 0$$
$$u_1(deviate|BB) = -c + \sum_{i=0}^{\infty} 0\delta^i = -c$$

Players are always better off to follow the system in the bad state.

Since both players start in the good state, grim-trigger system will make the cooperation if $\delta \geq c/b$. □

Given the discount factor δ is fixed, the ratio of cost c and benefit b directly make impact on the system performance. If $c = 0$, then $c/b = 0$ and every player will cooperate to relay the traffic from the opponent. Otherwise, if $b = 0$, then $c/b = \infty$ and there is no δ satisfying the equation, and all players do not have incentive to cooperate in this case. This is consistent with our intuition.

4.4.2 Tit-for-Tat System

Tit-for-tat system follows the human analogy "an eye for an eye, a tooth for a tooth". Every player will repeat whatever the opponent played in the previous round, as described in

Figure 4.3. If both of them cooperate, they will enjoy the cooperation forever. However, if one of them deviates from the responsibility, the opponent will deviate in the next round. In this system, punishment will result another punishment, which makes an infinite chain of "revenges".

Because of this chain, it is likely that the system becomes unstable. As proved in the following proposition, only a limited player will follow the rule, which shows that the system is not tolerant. An advantage is that it does not require explicit state maintenance because players repeat whatever they got.

Proposition 2 *Tit-for-Tat System induces the cooperation of the players as a subgame perfect Nash equilibrium if* $\delta = c/b$.

Proof: Suppose both players are in the good state. Then the rule is to play (R, R) *in the next game. If player 1 obeys the system, both players will stay in the good state. If player 1 deviates from the rule* (N, R)*, there will be infinite chain of revenge.*

$$\begin{aligned} u_1(obey|GG) &= \sum_{i=0}^{\infty}(b-c)\delta^i = \frac{b-c}{1-\delta} \\ u_1(deviate|GG) &= \sum_{i=0}^{\infty}(b-c\delta)\delta^{2i} = \frac{b-c\delta}{1-\delta^2} \end{aligned}$$

Player 1 will obey the rule if $\delta \geq c/b$ *in the* $(Good, Good)$ *state. The same argument applies to player 2.*

Suppose player 1 is in the good state but player 2 is the bad state. Then the rule is to play (N, R) *in the next round. Player 1 will stay in the revenge chain if he obeys the rule. If he deviates and forgives the opponent player, then they will enjoy the cooperation*

thereafter.

$$\begin{aligned} u_1(obey|GB) &= \sum_{i=0}^{\infty}(b-c\delta)\delta^{2i} = \frac{b-c\delta}{1-\delta^2} \\ u_1(deviate|GB) &= \sum_{i=0}^{\infty}(b-c)\delta^{i} = \frac{b-c}{1-\delta} \end{aligned}$$

Player 1 will obey the rule if $\delta \leq c/b$ in the $(Good, Bad)$ state.

Now, let's look at the player 2's perspective. If player 2 obeys the rule, then they stay in the revenge chain. Otherwise, both players go to bad state in the next round.

$$\begin{aligned} u_2(obey|GB) &= \sum_{i=0}^{\infty}(-c+b\delta)\delta^{2i} = \frac{-c+b\delta}{1-\delta^2} \\ u_2(deviate|GB) &= \sum_{i=0}^{\infty}0\delta^{i} = 0 \end{aligned}$$

Player 2 will follow the rule if $\delta \geq c/b$ in the $(Good, Bad)$ state. We can have the similar argument for $(Bad, Good)$ state.

Lastly, considering the (Bad, Bad) state, the rule is to play (N, N). Both players do not have incentives to deviate from the system because they will stay in that state regardless of the cooperation.

Since both players start with the $(Good, Good)$ state, they will cooperate if $\delta \geq c/b$. However, if we consider all the states together, players will follow the rule if $\delta = c/b$. □

4.4.3 Punish-and-Reward System

In punish-and-reward system, we include the notion of "forgiveness" into the grim-trigger system. Figure 4.4 illustrates the state diagram of the system. Both players start with the good state like in the previous systems. They enjoy the cooperation until one of them defects. The player who does not relay the opponent's traffic goes to a bad state. It can only restore its good reputation by receiving a punishment. Here, the punishment is to relay the

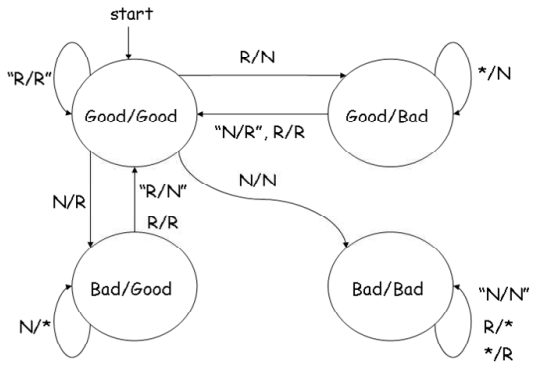

Figure 4.4: Punish-and-Reward State Diagram

opponent's traffic.

Since we have forgiveness here, the system is more flexible. In grim-trigger system, if there is a single mistake, then cooperation is impossible thereafter. However, this system allows the player to make up the mistakes.

In the following proposition, we analytically derive the condition that the players will follow the system.

Proposition 3 *Punish-and-Reward System induces the cooperation of players as a sub-game perfect Nash equilibrium if* $\delta \geq c/b$.

Proof: Suppose both players are in the good state. Then the rule is to play (R, R) *in the next game. If player 1 obeys the system, both players will stay in the good state. If player 1 deviates from the rule and plays* (N, R)*, then it will go to the* $(Bad, Good)$ *state. It should receive the punishment to get back to the* $(Good, Good)$ *state again.*

$$\begin{aligned} u_1(obey|GG) &= \sum_{i=0}^{\infty}(b-c)\delta^i = \frac{b-c}{1-\delta} \\ u_1(deviate|GG) &= b + \delta(-c) + \delta^2\frac{b-c}{1-\delta} \end{aligned}$$

Player 1 will obey the rule if $\delta \geq c/b$ *in the* $(Good, Good)$ *state. The same argument*

applies to player 2.

Suppose player 1 is in the good state but player 2 is the bad state. Then player 1 is supposed to play N and gets the reward. Player 1 will receive the reward from player 2 regardless of its action, and the players will go to $(Good, Good)$ state.

$$\begin{aligned} u_1(obey|GB) &= b + \delta\frac{b-c}{1-\delta} \\ u_1(deviate|GB) &= (b-c) + \delta\frac{b-c}{1-\delta} \end{aligned}$$

Player 1 is always better off to follow the rule.

Now, let's look at the player 2's perspective. Player 2 is supposed to get the punishment and play R. If player 2 obey the rule and receives the punishment (N, R), then it recovers its reputation. Otherwise, it deviates from the rule (N, N), it stays in the bad state and defers the punishment to later game.

$$\begin{aligned} u_2(obey|GB) &= (-c) + \sum_{i=1}^{\infty}(b-c)\delta^i = -c + \delta\frac{b-c}{1-\delta} \\ u_2(deviate|GB) &= 0 + \delta(-c) + \delta^2\frac{b-c}{1-\delta} \end{aligned}$$

Player 2 will follow the rule if $\delta \geq c/b$ in the $(Good, Bad)$ state. We can have the similar argument for $(Bad, Good)$ state.

Lastly, considering the (Bad, Bad) state, the rule is to play (N, N). Both players do not have incentives to deviate from the system because they will stay in that state regardless of the cooperation.

The system will successfully make the cooperation if $\delta \geq c/b$. □

From three propositions above, we can see that the conditions to make players cooperate are similar for all three candidates. However, punish-and-reward system is the most promising one because it is tolerant against some possible mistakes. In addition, it will guarantee the convergence of the overall overlay network.

4.4.4 Generalized Punish-and-Reward System

We can generalize the idea of punish-and-reward system. Instead of making an one-time punishment, we can think of M consecutive punishments as a make-up for the defecting. Intuitively speaking, we may enforce more players to cooperate as we increase the number of punishments. If the player refuses a punishment during the period, then the punishment procedure restarts from the beginning.

The following proposition shows that the system becomes more tolerant against impatient players as we increase the number of punishments.

Proposition 4 *Generalized Punish-and-Reward System induces the cooperation of players as a subgame perfect Nash equilibrium if* $\sum_{i=K}^{M} \delta^i \geq c/b$ *for* $1 \leq K \leq M$.

Proof: Suppose both players are in the good state. Then the rule is to play (R, R) *in the next game. If player 1 obeys the system, both players will stay in the good state. If player 1 deviates from the rule* (N, R)*, then it will go to the* $(Bad, Good)$ *state. It should receive the* N *punishments to get back to the* $(Good, Good)$ *state again.*

$$\begin{aligned} u_1(obey|GG) &= \frac{b-c}{1-\delta} \\ u_1(deviate|GG) &= b + \sum_{i=1}^{M} (-c)\delta^i + \delta^{M+1}\frac{b-c}{1-\delta} \end{aligned}$$

Player 1 will obey the rule if $\sum_{i=1}^{M} \delta^i \geq c/b$ *in the* $(Good, Good)$ *state. The same argument applies to player 2.*

Suppose player 1 is in the good state but player 2 is the bad state. Let K *be the number of punishments left for player 2. Then player 1 is supposed to play* N *and gets the*

reward. Player 1 will receive the reward from player 2 regardless of its action.

$$\begin{aligned} u_1(obey|GB_K) &= \sum_{i=0}^{K-1} b\delta^i + \delta^K \frac{b-c}{1-\delta} \\ u_1(deviate|GB_K) &= (b-c) + \sum_{i=1}^{K-1} b\delta^i + \delta^K \frac{b-c}{1-\delta} \end{aligned}$$

Player 1 is always better off to follow the rule.

Now, let's look at the player 2's perspective. Player 2 is supposed to get the K more punishments and play R. If player 2 obey the rule and receives the punishment (N, R), then it recovers its reputation. Otherwise, it deviates from the rule (N, N), it stays in the bad state and start the punishment process from the beginning.

$$\begin{aligned} u_2(obey|GB_K) &= \sum_{i=0}^{K-1} (-c)\delta^i + \delta^K \frac{b-c}{1-\delta} \\ u_2(deviate|GB_K) &= 0 + \sum_{i=1}^{N} (-c)\delta^i + \delta^{N+1} \frac{b-c}{1-\delta} \end{aligned}$$

Player 2 will follow the rule if $\sum_{i=K}^{M} \delta^i \geq c/b$ in the $(Good, Bad_K)$ state. We can have the similar argument for $(Bad_K, Good)$ state.

Lastly, considering the (Bad, Bad) state, the rule is to play (N, N). Both players do not have incentive to deviate from the system because they will stay in that state regardless of the cooperation.

The system will successfully make the cooperation if $\sum_{i=1}^{M} \delta^i \geq c/b$. □

4.5 Simulation

In this section, we conduct simulation to evaluate one of our proposed methods, punish-and-reward system. First, we want to check if the system actually encourages much cooperation. Also, it is important to check the *price of anarchy*, a metric to measure how much the system

is close to the social optimum.

4.5.1 Parameters

We have three main parameters to set in the simulation:

1. Benefit and cost values: Given a fixed network topology and traffic demand, we may calculate the benefit value using network coordinate systems. However, we have not implemented this feature. For this simulation, we teak the maximum value of benefit and costs, and use uniform distribution to allocate these values for each pair of the overlay nodes.

2. Number of punishments: We change the number of punishments to see the impact on the cooperation ratio.

3. Discount factor: In order to test the tolerance of the system, we gradually decrease the discount factor.

4.5.2 Evaluation Metrics

We want to evaluate the system in terms of two factors: cooperation ratio and overall performance.

1. Degree of Cooperation (DoC): This metric simply counts the number of cooperations after the system converges. Clearly, we want to make this metric as high as possible.

$$\text{DoC} = \frac{\text{Number of Cooperations}}{\text{Number of Possible Pairs of Players}}$$

2. Price of Anarchy (PoA): We want to quantify the performance degrading by the absence of central control, which is called *price of anarchy*.

$$\text{PoA} = \frac{\text{Optimal Performance}}{\text{Actual Performance}}$$

If the performance is close to optimal, it gets close to 1. The value gets closer to ∞ if the

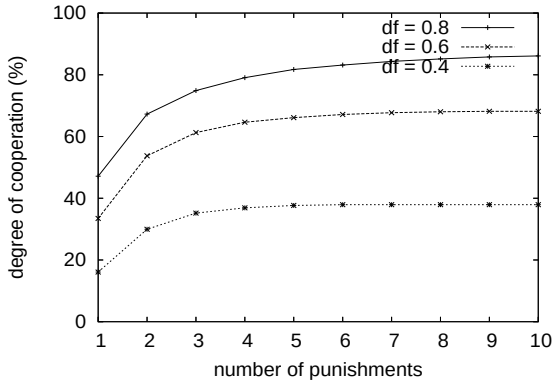

Figure 4.5: As we put more punishments, more players tend to cooperate. The degree of cooperation converges at some level.

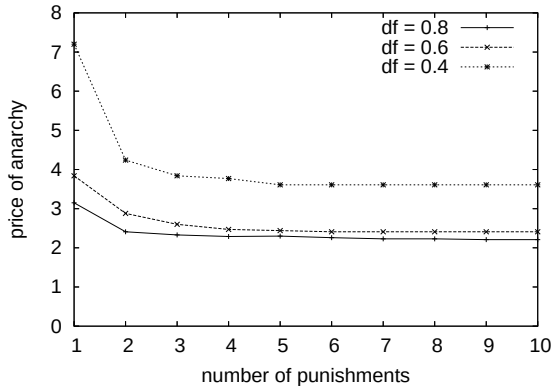

Figure 4.6: More punishment improves the overall system optimality. Especially, the second punishment significantly improves the performance.

actual performance gets worse.

Another possible metric is fairness. We want the regulation of our system affects the players as equally as possible. In other words, we want the ratio between actual benefit and actual cost to be as uniform as possible for each overlay node. We will include this metric in the future work.

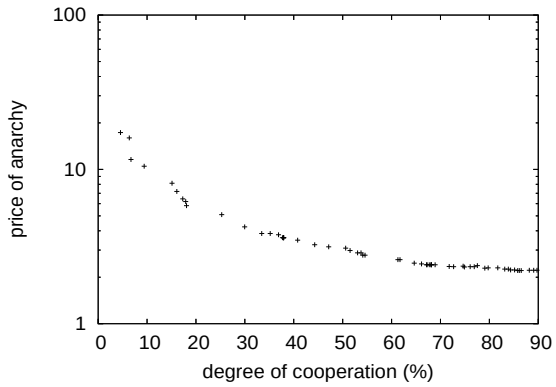

Figure 4.7: Price of anarchy and degree of cooperation have strong correlation. More cooperation induces better performance.

4.5.3 Simulation Result

The first question we want to answer is: *Does the system make incentives to cooperate?* In Figure 4.5, we plot the degree of cooperation as a function of the number of punishments. Following our intuition, more players tend to cooperate as we increase the period of punishments. Interesting thing is that the degree of cooperation converges with a limited number of punishments. It seems that we do not need to that many punishments to stimulate most players. Regardless of the discount factor values, the degree of cooperation converges when we have more than four punishments.

As the next experiment, we test the price of anarchy as we increase the number of punishments. As shown in Figure 4.6, the price of anarchy converges, which is similar to the case of the degree of cooperation.

After the two experiments, we come up with a question: *Is cooperation always good for the social performance?* In Figure 4.7, we try to find the correlation between the price of anarchy and the degree of cooperation. The simulation shows that there is strong correlation between those factors. Therefore, we can say that encouraging more cooperation will also improve the social performance.

Lastly, we want to check the tolerance of the system as we decrease the discount

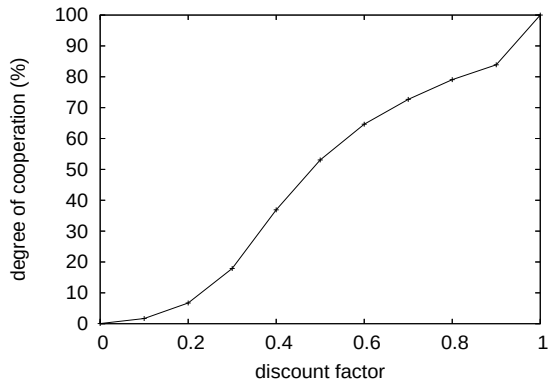

Figure 4.8: The system shows tolerance to "moderate" players.

factor (Figure 4.8). The system shows tolerance for moderate players ($\delta \geq 0.4$). However, as players become extremely impatient ($\delta \leq 0.3$), we could not get good degree of cooperation.

4.6 Related Work

Incentive issues have been extensively researched in networked systems. Especially, incentives in P2P files sharing systems are distinguishing [10, 12, 20, 26, 30]. Most studies on P2P file sharing system focus on the achievement of optimality by preventing free-riders, who do not contribute to the system. In [12, 26], non-cooperative game is used to explain self-interested users in P2P networks.

There are mainly two methods to incentivize the cooperation: monetary payments and differential services. Afergan and Wroclawski [7] introduces explicit incentives by monetary system into routing. However, the monetary system seems highly impractical in network society [31]. Thus, [20, 26] proposes differential service model for P2P systems. In their model, service level users can achieve is proportional to the contribution.

Recently, there have been some works on economic issues on overlay routing. Chun et al. [15] characterize selfishly constructed overlay routing networks and prove

that selfishly characterized overlay networks can present desirable properties with a non-cooperative game model. In [13], a cost-based model is adopted to assess the network resources in overlay networks and the benefits of participating in the overlay networks are explained as a cost reduction. In [40], market-driven approach to regulate selfish nodes is proposed for bandwidth allocation problem in overlay networks.

In most of the previous works, the interaction of networked systems is modeled as an one-shot game. However, repeatability should be considered in overlay routing because overlay network consists of well-identified routers, and the interaction among them is continuous. Afergan and Sami [6] study multicast application overlay networks in a repeated-game framework. In their paper, users have both the motivation and the means to alter their position in the overlay tree. Our work is different from it in the sense that the users' action space is to selectively relay transit traffic, instead of moving their position in overlay network.

4.7 Conclusion and Future Work

In this chapter, we analyze the overlay routing in the assumption that the overlay nodes only cooperate each other if there is an incentive to do so. We model the individual overlay nodes as strategic and rational players to maximize their own utility. The game-theoretic analysis shows that we need some mechanism to encourage the cooperation. We propose three trigger-based frameworks to solve this problem with analytic proof for their feasibility. With simulation, we show the feasibility of the generalized punish-and-reward system.

In the process of approaching the problem, we introduce some bold assumptions to be released. First, we assume that the overlay node can forward as much traffic as it wants. However, in reality, every router has some limitation in network bandwidth and

computation power. Therefore, we need to add a constraint to the problem.

$$\begin{aligned} \max \quad & \sum_{i \neq j} u_{i,j} \\ \text{subject to} \quad & \sum_{i \neq j} c_{i,j} \leq \Theta \end{aligned}$$

where Θ is the maximum cost an overlay node can afford.

Another assumption is about the benefit. We have not modeled how these values are set up. Since the network changes dynamically, the benefit values will be changed by the new traffic flow. Actually, the value may change by the overlay traffic by itself because an attractive overlay node may cause congestion to the neighbor links.

Basically, this chapter focuses on the *micro-level* interaction inside of a single overlay, whereas the previous two chapters study the *macro-level* interactions. Then an interesting topic is to generalize the incentive-based idea to the macro-level overlay relations. It would be interesting to analyze the impact of internal interaction to the external interaction and the opposite direction as well.

Chapter 5

Conclusion

This thesis studies three interactions related to overlay routing. Overlay routing allows end hosts to make routing decisions at the application level. By its underlying definition, overlay determines the logical routing to optimize its own performance, and this selfishness affects the performance of the related components.

First, we improve the *vertical interaction* between overlay routing and traffic engineering by modifying the objectives of both parties. Overlay changes the physical traffic demands by the logical routing. The dynamically changing traffic demands affect the performance of underlay routing. Specifically, network operators use traffic engineering techniques to adapt IP layer routing to cope with the new traffic matrix. We propose *TE-aware overlay routing*, which takes traffic engineering's objective into account. Then we also suggest COPE as a strong traffic engineering technique, which makes a good interaction with the unpredictable overlay traffic demands. We show the feasibility of the proposed methods with extensive simulation results.

Secondly, we show the initial results on the *horizontal interaction* among multiple overlays coexisting on top of a shared underlay networks. The decision of an overlay depends on the probing information for every logical path. If overlays share some links but do not exchange the routing information of each other, currently observed link performance is

likely be changed by actions of other overlays. When the underlay routing is static, the *pure* horizontal interactions do not make oscillation issues. However, when traffic engineering dynamically changes its underlay routing, the simulations show that MPLS has poor interaction with multiple overlays. We again show that TE-awareness and demand-obliviousness improve the interaction.

Lastly, we propose an incentive-based framework for *internal interaction* inside an overlay. We model the individual overlay nodes as strategic decision makers to maximize their payoffs. Each overlay node selectively forward transit traffic from other overlay nodes to maximize the benefits from overlay routing and to minimize the cost to relay transit traffic. We formulate the internal interaction as a non-cooperative game between overlay nodes. We apply repeated game theory to capture the repeated feature of overlay routing. With the analytic arguments, we show the condition in which the proposed method works efficiently. Then we describe the simulation results to support our argument.

Throughout the thesis, we analyze the interactions of overlay routing with game theory framework. Many networks and distributed systems involve with strategic interaction between multiple independent components, who seek individual gain from the system. Then we can use game-based approach to analyze the selfish behavior of the individuals and design incentive-based frameworks to achieve overall system optimality.

Bibliography

[1] Akamai Technologies, Inc. http://www.akamai.com.

[2] Condor. http://www.cs.wisc.edu/condor.

[3] General Algebraic Modeling System. http://www.gams.com.

[4] Gnutella. http://www.gnutella.com.

[5] OSPFv3 overview. 2003.

[6] M. Afergan and R. Sami. Repeated-game modeling of multicast overlays. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, April 2006.

[7] M. Afergan and J. Wroclawski. On the benefits and feasibility of incentive based routing infrastructure. In *Proceedings of ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems (PINS)*, pages 197–204, 2004.

[8] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceedings of Symposium on Operating Systems Principles (SOSP)*, pages 131–145, 2001.

[9] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain trafficc demands: Understanding fundamental tradeoffs. In *Proceedings of ACM SIGCOMM*, pages 313–324, 2003.

[10] A. Blanc, A. Vahdat, and Y.-K. Liu. Designing incentives for peer-to-peer routing. In *Proceedings of the Second Workshop on Economics of Peer-to-Peer Systems*, 2004.

[11] R. Braden, D. Clark, and S. Shenker. Integrated services in the Internet architecture: an overview. Internet Request for Comment RFC 1633, Internet Engineering Task Force, June 1994.

[12] C. Buragohain, D. Agrawal, and S. Suri. A game theoretic framework for incentives in P2P systems. In *Proceedings of IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2003.

[13] N. Christin and J. Chuang. A cost-based analysis of overlay routing geometries. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, March 2005.

[14] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling conferencing applications on the internet using an overlay muilticast architecture. In *Proceedings of ACM SIGCOMM*, pages 55–67, 2001.

[15] B.-G. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz. Characterizing selfishly constructed overlay routing networks. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2004.

[16] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of ACM SIGCOMM*, 2004.

[17] P. K. Dutta. Strategies and games: Theory and practice, 1999.

[18] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine*, pages 118–124, October 2002.

[19] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In

Proceedings of IEEE International Conference of Computer Communications (INFOCOM), pages 519–528, 2000.

[20] P. Golle, K. Leyton-Brown, I. Mironov, and M. Lillibridge. Incentives for sharing in peer-to-peer networks. *Lecture Notes in Computer Science*, 2001.

[21] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. O. Jr. Overcast: Reliable multicasting with an overlay network. In *Proceedings of Symposium on Operating Systems Design and Implementation (OSDI)*, pages 197–212, 2000.

[22] W. Jiang, D.-M. Chiu, and J. C. Lui. On the interaction of multiple overlay routing. *Journal of Performance Evaluation*, pages 335–350, October 2005.

[23] R. Keralapura, C.-N. Chuah, N. Taft, and G. Iannacco. Can coexisting overlays inadvertently step on each other? In *Proceedings of International Conference on Network Protocols (ICNP)*, pages 201–214, 2005.

[24] R. Keralapura, N. Taft, C.-N. Chuah, and G. Iannacco. Can ISPs take the heat from overlay networks? In *Proceedings of the Third Workshop on Hot Topics in Networks*, November 2004.

[25] P. B. Key and D. McAuley. Differential QoS and pricing in networks: Where flow control meets game theory. *IEE Proceedings - Software*, 146(1):39–43, 1999.

[26] L. Lai, M. Feldman, I. Stoica, and J. Chuang. Incentives for cooperation in peer-to-peer networks. In *Proceedings of the First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, 2003.

[27] Y. Liu, H. Zhang, W. Gong, and D. Towsley. On the interaction between overlay routing and traffic engineering. In *IEEE Conference on Computer Communications (INFOCOM)*, 2005.

[28] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: existing techniques and new directions. In *Proceedings of ACM SIGCOMM*, pages 161–174, 2002.

[29] J. Nash. Non-cooperative games. *The Annals of Mathematics*, pages 286–295, September 1951.

[30] T.-W. Ngan, A. Nandi, A. Singh, D. S. Wallach, and P. Druschel. On designing incentives-compatible peer-to-peer systems. In *Proceedings of the Second Workshop on Future Directions in Distributed Computing*, 2004.

[31] A. Odlyzko. The history of communications and its implications for the Internet. Manuscript, June 2000.

[32] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker. On selfish routing in Internet-like environments. In *Proceedings of ACM SIGCOMM*, pages 151–162, Karlsruhe, Germany, august 2003.

[33] J. Ruela and M. Ricardo. MPLS - Multi-Protocol Label Switching. In *The Industrial Information Technology Handbook*, pages 1–9. 2005.

[34] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: a case for informed internet routing and transport. Technical Report TR-98-10-05, 1998.

[35] M. Seshadri and R. H. Katz. Dynamics of simultaneous overlay network routing. Technical Report UCB/CSD-03-1291, EECS Department, University of California, Berkeley, 2003.

[36] J. W. Stewart, III. *BGP4: inter-domain routing in the Internet*. Addison-Wesley, 1999.

[37] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. OverQoS: An overlay

based architecture for enhancing internet QoS. In *Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI)*, pages 71–84, 2004.

[38] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The impact of routing policy on internet paths. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pages 736–742, 2001.

[39] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg. COPE: Traffic engineering in dynamic networks. In *Proceedings of ACM SIGCOMM*, 2006.

[40] W. Wang and B. Li. Market-driven bandwidth allocation in selfish overlay networks. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, March 2005.

Vita

Gene Moo Lee was born in Seattle, Washington on July 7, 1981, the son of Dr. Sang-Won Lee and Mrs. Kyung-Ja Hwang. After completing his work at Daejeon Foreign Language High School, Daejeon, Korea, in 2000, he entered Korea University in Seoul, Korea. He received two Bachelor's degrees from Korea University in February, 2004. The first was a Bachelor of Science in Computer Science and Engineering, and the second was a Bachelor of Science in Mathematics. In August, 2004, he joined the graduate program in Department of Computer Sciences at The University of Texas at Austin. During the summer of 2005, he was employed as an analyst at Goldman Sachs Execution & Clearing, L.P., New York, New York. As of July 2006, he works as a researcher at Samsung Advanced Institute of Technology, Gyunggi, Korea.

Permanent Address: Hanbit Apt. 131-204, Eo-eun Dong
Yuseong Gu, Daejeon, 305-755, Republic of Korea
Email: gene@cs.utexas.edu

This thesis was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society.