Practical and Efficient Internet Routing with Competing Interests

Ratul Mahajan

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2005

Program Authorized to Offer Degree: Computer Science and Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Ratul Mahajan

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Co-Chairs of the Supervisory Committee:

____
David J. Wetherall

____
Thomas E. Anderson

Reading Committee:

____
David J. Wetherall

____
Thomas E. Anderson

____
John Zahorjan

Date: ______________________

University of Washington

Abstract

Practical and Efficient Internet Routing with Competing Interests

Ratul Mahajan

Co-Chairs of the Supervisory Committee:
Associate Professor David J. Wetherall
Computer Science and Engineering

Professor Thomas E. Anderson
Computer Science and Engineering

A fundamental characteristic of the Internet, and increasingly other networked systems, is that it is controlled by many independent parties that act in their own interests. These parties, known as Internet service providers (ISPs), cooperate to provide global connectivity even though they often have competing interests. With current routing protocols, this competition induces each ISP to select paths that are optimal within its own network. The result is that paths across multiple networks can be poor from a global perspective.

I present Wiser, a protocol that shows it is possible to achieve efficient global routing in practice even when ISPs select paths in their own interests. Wiser is based on "barter" between pairs of adjacent ISPs: each ISP selects paths that respect the concerns of the neighboring ISP as well as its own in return for the neighbor doing the same. To encourage ISPs to adopt Wiser, it is designed to maintain their autonomy, e.g., it does not require sensitive internal information to be disclosed, and individual ISPs do not lose compared to routing in the Internet today. Wiser can be implemented as an extension to current Internet routing protocols and deployed incrementally.

To evaluate Wiser, I experiment with measured ISP topologies and a router-level prototype. I find that, unlike Internet routing today, the efficiency of Wiser is close to that of an ideal routing that globally optimizes network paths for metrics such as path length or bandwidth provisioning. I further show that these benefits come at a low cost: the overhead of Wiser is similar to that of the current Internet routing protocol in terms of routing messages and computation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Number | Page

ACKNOWLEDGMENTS

I was very fortunate to have a great advisory team in David Wetherall and Tom Anderson, who whetted my appetite for computer science research. As I start my research career, I hope that I can emulate, at least partly, David's keen eye for the big picture and love for detail, and Tom's intrepidity.

I have learned much from my other mentors and collaborators as well, for which I am grateful. Neil Spring is smart, unassuming and perfectionistic. To this day, his methods simplify many of my tasks. John Zahorjan has remarkable insights into most situations, which are surpassed only by his willingness to help others see the point. With their enthusiasm, Maya Rodrig and Charles Reis gave me a chance to explore new research directions. Sushant Jain's intensity exceeds that of most people I know. Sally Floyd's dedication to Internet research is inspirational; for her, it is personal. Miguel Castro and Antony Rowstron showed me that, given the right colleagues, work can be fun.

I thank my officemates over the years – Tammy VanDeGrift, Igor Tatarinov, Sarah Peterson, Jenny Liu, Michael Cafarella, Markus Mock, Alex Mohr, and Yaw Anokwa – for humoring me whenever I was distracted.

Several fellow students, including Seth Bridges, Mira Dontcheva, David Grimes, Wilmot Li, and Maya Rodrig ensured that I had a life outside of work.

I am thankful to Cisco Systems, Microsoft Research, and the NSF for funding the work presented in this dissertation.

Lastly, and most importantly, I am indebted to my parents for their support and having faith in me to make the right choices, well, at least after I got to a certain age.

Chapter 1

INTRODUCTION

> The most important change in the Internet architecture over the next few years will probably be the development of a new generation of tools for management of resources in the context of multiple administrations.
>
> David Clark, 1988
>
> *Design Philosophy of the DARPA Internet Protocols*

A defining characteristic of many large-scale distributed systems is that parts of them are controlled by independent parties that must cooperate to provide a useful service. But often these parties have competing interests, i.e., their interests are not completely aligned with each other. The reasons for this non-alignment include, but are not limited to, direct economic competition between the parties. Examples of such distributed systems include the Internet and email, in which global connectivity is provided by competing providers, and peer-to-peer and wireless networks that have users who are more interested in consuming rather than contributing resources [2, 104].

Competition induces these parties to hide information and make selfish decisions, which can degrade the efficiency and the stability of the system [2, 71, 87, 109]. For instance, a common practice in the Internet today is *early-exit* routing, in which the Internet service providers (ISPs) transfer packets to their neighboring ISPs at an interconnection that is closest to the point of entry in their own network without regard to the destination of the packets inside the neighbor's network. While this minimizes the resources consumed in the upstream ISP's network, it can inflate the length of the end-to-end path of these packets [109].

Protocols for competitive environments must ensure efficient and robust operation in the face of selfish behavior by the constituent parties. The pervasiveness of competition today makes designing such protocols an important research agenda, and one that spans distributed systems, economics, theoretical computer science, and sociology [39, 38, 89, 87, 49]. But designing such protocols is challenging. The protocols must reflect the interests of individual parties and be robust against strategic behavior which can include deviation from the protocol specification itself. At the same time, they must be scalable and easy to implement and deploy. Perhaps because of these challenges, there are few successful examples of such protocols, even though their need was articulated by Clark seventeen years ago (see the quote above) [29]. Their continued scarcity is underscored by the fact that he reiterated it recently [30]: "The challenge facing Internet research and engineering is to recognize and leverage [the presence of competing interests] – at minimum to accommodate it."

In this dissertation, I address this challenge in the context of Internet routing, with the hope that lessons from this concrete example will be applicable to a broader class of competitive-yet-cooperative systems. The Internet consists of thousands of ISPs, none of which has a global reach. Cooperating by carrying data for one another is how they provide global connectivity to their customers. But often these ISPs also compete with one another as business entities, for instance, for the same set of customers. This motivates the ISPs to be selfish. Because of the way in which this selfishness manifests itself today, it degrades the efficiency of routing paths and the performance of applications that use the Internet.

My solution is a practical protocol, called Wiser, in which pairs of ISPs barter with each other for mutual gain. I use Wiser to demonstrate that, even when ISPs continue to act in their own interests, it is possible to achieve efficient routing that is close to an ideal sys-

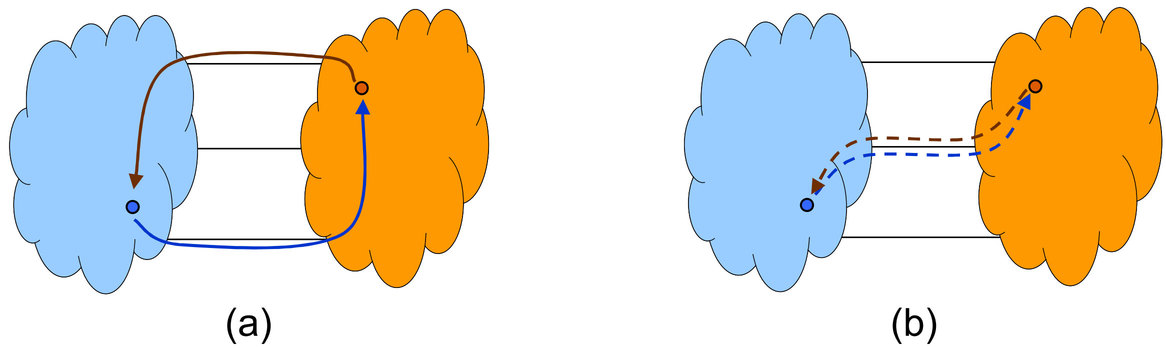


Figure 1.1: Locally optimal path selection leads to longer paths as well as higher costs for ISPs. Each cloud represents an ISP and the horizontal lines represent the interconnections between them. (a) Early-exit routing: each ISP selects a locally optimal interconnection for traffic leaving its network. (b) A routing pattern that has shorter paths and is better for both ISPs when considering traffic flowing in both directions.

tem that is globally optimized. Below, I discuss the problems with Internet routing today, requirements that a practical solution must satisfy, and why existing proposals are insufficient. I then provide an overview of my approach and solution, including its contributions.

1.1 Internet Routing Today

Routing in the Internet today operates at two logically distinct levels. *Intradomain* routing is used by ISPs to reach destinations inside their own networks. *Interdomain* routing, which is the subject of this thesis, is used by ISPs to reach destinations inside other ISPs' networks. As part of interdomain routing, neighboring ISPs carry traffic for each other based on bilateral contracts.

ISPs tend to select locally optimal interdomain routing paths because the cost that an ISP incurs for carrying a packet depends on the path inside the ISP and because ISPs have limited knowledge about other ISPs' networks. Thus, for each packet that an ISP controls, it selects a path that minimizes the local cost, largely disregarding the cost incurred by other ISPs. This can lead to inefficient and unstable routing paths. Paths can be inefficient because locally optimal routing decisions can be poor from a global standpoint [109, 105, 118]. For instance, early-exit routing, shown in Figure 1.1, leads to longer than necessary paths. Longer paths are undesirable because they consume more resources in the network and the performance of most applications is inversely correlated with path length.

Paths can be unstable because the actions of ISPs influence each other. Since an ISP's path selection is based largely on local concerns, it can adversely impact another ISP, for instance, by causing congestion. In some cases, the other ISP will react by changing its own path selections, and in the worst case, cycles that lead to long-term instability can arise. One such incident that I am aware of involved two large ISPs and lasted for two days [46].

Lacking protocol support to address inefficiencies and instabilities that arise in practice, ISPs try to reduce the occurrence of serious problems through network engineering. For instance, to minimize the ill-effects of early-exit routing, they interconnect widely. While this makes the common case acceptable, it does not eliminate all problems. To fix the remaining problems, ISPs rely on "tweak and pray" [37, 10], or manual re-configuration of routing (as was the case in the incident above). Manual control of routing is unreliable. For instance, one study found that more than three quarters of new routing messages result from configuration errors [73]. Manual control also increases the cost of operating an ISP network, which is ultimately borne by the end users.

In addition to poor efficiency and reliability, current Internet routing also leads to a higher than necessary cost of carrying traffic for all ISPs. For instance, early-exit routing in Figure 1.1(a) is more expensive for both ISPs compared to the routing in Figure 1.1(b) because the total cost to carry the two flows incurred by each ISP is more. Even so, ISPs continue to use locally optimal paths because the current protocols provide no incentive to move away from this behavior – the cost of carrying traffic can be higher for an ISP when it does not select locally optimal paths while other ISPs continue to do so. I believe that, given protocols that preserve their interests, ISPs will cooperate to improve routing. This is evidenced in their use of ad hoc manual cooperation today to fix problems. Reflecting ISPs' interests is therefore central to the design of the routing protocol presented in this dissertation.

1.2 Goal

My goal is to design a practical protocol that enables ISPs to automatically discover efficient routing paths. Being practical requires that the overhead of the protocol, in terms of factors such as implementation and routing message complexity, not be much higher than that of the currently deployed protocols. It also requires that the protocol preserve the autonomy of individual ISPs, which implies the following in the context of Internet routing. First, ISPs should not be required to disclose sensitive internal information about their networks, such as monetary cost, topology or performance, because ISPs are reluctant to share this information. This reluctance stems from competitive concerns, i.e., fear that a competitor might use this information to its advantage. Second, ISPs should be able to reflect diverse optimization criteria, because different ISPs optimize their networks differently. For instance, some ISPs minimize path length and some minimize link utilization. Third, individual ISPs should not lose compared to the routing today. I refer to this property as *win-win*, and in its absence, ISPs that lose will have little incentive to adopt any new protocol.

The conceptual framework in Figure 1.2 illustrates my goal. It shows the extent to which parties make local decisions on the x-axis and the routing quality, measured in any unit of interest, on the y-axis. I coarsely divide the x-axis into two halves. In the right half, ISPs make decisions in social interest. In the left half, ISPs make autonomous decisions in self-interest. Consider the two extreme operating modes on the x-axis. On the right is *anarchy*, in which the competing parties route data based on self-interest and without any coordination with other parties. While today's routing is not a complete anarchy because ISPs operate within a contractual framework, it is close to being one, especially in the way most routing paths are selected [109]. Anarchy is undesirable if it leads to poor routing

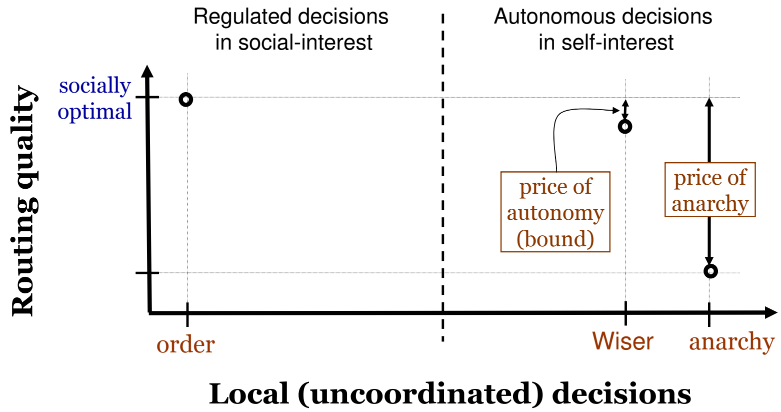


Figure 1.2: A conceptual framework for routing between independent parties.

quality. The other extreme is *order*, in which the parties route data based purely on social interests, to globally optimize routing quality. This operating mode completely ignores the interests of individual parties, and as such, it will emerge only in a heavily regulated network. The difference in the quality of routing between order and anarchy is known as the *price of anarchy* [87]. Low quality of anarchic routing leads to a high price of anarchy.

However, anarchy is not the only operating mode that preserves ISPs' autonomy. In particular, ISPs can coordinate if it meets all of the requirements specified above. Such coordination is self-enforcing in that it does not require regulation to be sustainable [9]. The *price of autonomy* can be similarly defined as the difference in the quality of routing between order and autonomous coordination. The goal of this work, depicted by "Wiser" in the figure, is to design a protocol that achieves high routing quality with low-overhead coordination that preserves ISP autonomy. The efficiency of any concrete routing protocol provides an upper bound on the price of autonomy in the Internet.

1.3 Existing Solutions

Existing protocols to improve the efficiency of Internet routing are insufficient to address the challenges posed by a competitive environment. I review existing work in detail in Chapter 7, but it can be divided into two categories. The first category is composed of techniques that individual ISPs can use without coordination with others to improve their own traffic [119, 91, 37, 120]. However, such techniques do not encourage ISPs to stop selecting locally optimal paths in favor of paths that are better from a global perspective. An ISP that stops selecting locally optimal paths stands to lose because, in the absence of any coordination, it is not guaranteed that other ISPs will reciprocate. The failures of these techniques to extricate ISPs from the lose-lose situation of anarchy implies that they have limited effectiveness in practice [90]

The second category is that of techniques that rely on ISP coordination [3, 79, 38, 69, 127, 128]. However, their coordination mechanisms do not reflect one or more of the ISPs' interests mentioned above. Some require ISPs to disclose sensitive information such as the monetary cost of carrying traffic [38, 79, 3]; some do not enable ISPs with diverse objectives to coordinate their routing and instead assume that all ISPs have the same optimization criteria [69, 128]; and some do not lead to win-win routing [69, 128]. As a result, these techniques face significant adoption hurdles [53].

1.4 My Approach

My approach to designing a practical and efficient routing protocol is based on *bilateral barter* between adjacent ISPs and *agnostic costs*. With bilateral barter, an ISP does a favor to its neighboring ISP by selecting paths that account for both the local cost as well as the cost incurred by the neighbor; the neighbor returns the favor when it selects paths for the traffic that it sends to this ISP. Unlike routing today, bilateral barter encourages ISPs to reflect global costs while selecting paths. For instance, it can lead to the routing pattern in Figure 1.1(b): the left ISP does a favor to the right ISP by using the middle interconnection, instead of the bottom one, for traffic leaving its network, and the right ISP returns the favor for traffic leaving its network. The insight underlying barter is that when ISPs take a holistic view of the traffic they exchange, their interests are not completely opposed to each other. Optimizing individual paths can lead to a gain for one ISP and a loss for another, but systematically optimizing across the entire set of paths can benefit both ISPs because the loss from optimizing an individual path is usually smaller than the gain. Even though bilateral barter focuses on improving the situation for individual ISPs, I show that it leads to better end-to-end paths as well.

Agnostic costs form the basis of information sharing between ISPs, which is required to implement barter so that one ISP knows what paths are preferred by the other. These costs represent cardinal preferences – whose relative magnitude is of significance – of an ISP for various paths within its network. They are derived from the ISPs' internal optimization criterion such as minimizing resource usage. With agnostic costs, ISPs are not required to directly disclose sensitive information about their network, and ISPs can control how much information is disclosed. ISPs are not required to disclose either the optimization criterion or the derivation methodology. Agnostic costs also enable ISPs with diverse objectives to coordinate. Cardinal preferences are more expressive than ordinal preferences – whose ordering but not relative magnitude is of significance – which ISPs disclose today as part of a commonly used routing mechanism. I show that, unlike ordinal preferences, cardinal preferences provide an effective basis for efficient routing.

1.5 My Solution: Wiser

In this dissertation, I present Wiser, a practical routing protocol based on bilateral barter and agnostic costs. With Wiser, downstream ISPs disclose their agnostic costs to upstream ISPs as part of their routing announcements, and the upstream ISPs select routing paths based on both their internal costs and the downstream ISPs' costs. The costs are weighted such that the resulting paths are sensitive to the concerns of both ISPs, rather than being dominated by one of them. I build on existing off-line contracts between neighboring ISPs to make Wiser less vulnerable to strategic manipulation, by restricting the degrees of freedom available to ISPs. These restrictions stem from normalizing ISPs' costs to satisfy a certain criterion and placing a contractual limit on the average cost an ISP incurs for carrying traffic received from another ISP.

I evaluate the efficiency, overhead and robustness to cheating of Wiser using measured ISP topologies and realistic workloads. I study efficiency in scenarios with both comparable and incomparable ISPs' objectives. For comparable objectives, I consider two metrics motivated by current problems with the Internet routing [109, 62]. The first metric is the length of Internet paths, which directly impacts application performance. The second metric is a measure of the amount of bandwidth provisioning ISPs require to deal with load variations, e.g., variations due to failures. For both metrics, Wiser is more efficient than *anarchy* which I model using current routing practices, and its efficiency is close to optimal routing, a hypothetical scenario in which the routing is globally optimized using complete information. Compared to optimal routing, the average path length is only 4% higher with Wiser and 13% higher with *anarchy*. While this average gain is useful, the key difference is in the tail of the path length distribution. The worst 1% of the paths are 6 times longer with *anarchy* but only 1.5 times longer with Wiser. For the bandwidth metric, Wiser reduces ISP network provisioning requirements by 8% on average compared to *anarchy*. With incomparable ISPs' objectives, I find that Wiser enables ISPs to cooperate such that each gains according to its own objective, and efficient end-to-end paths are obtained when ISP objectives are partially grounded in metrics of interest to end users. For the cheating strategies that I study, I show that Wiser limits the gains for dishonest ISPs and the losses for honest ISPs.

I implement Wiser in XORP [129], an experimental router platform, and in SSFNet [113], a network simulator, to study its overhead along several dimensions. I find that it is easy to implement: starting from implementations of the current routing protocol, it requires less than 6% additional lines of code. The overhead due to routing messages generated by Wiser is similar to that of the current interdomain routing protocol. For normal routing workloads that routers experience today, the computation overhead of Wiser is within 15 to 25% of the current protocol.

My results are of course limited to the topologies, workloads and ISPs' behaviors that I study. They are intended to be indicative of what will happen in a real deployment.

As a tool developed for strategic parties, Wiser provides a framework within which ISPs can cooperate, but how they chose to do so is hard to predict in advance [30]. In particular, Wiser produces efficient routing when ISPs disclose agnostic costs that reflect their objectives and when they select paths as recommended by the protocol. While I argue that ISPs have an incentive to do so, whether a particular ISP chooses to depends on that ISP. Similarly, while Wiser will likely lead to stable routing if ISPs follow certain simple guidelines and network engineering practices, it does not guarantee stability under all possible scenarios. (This is similar to today's interdomain routing protocol, and the stability of routing in large networks, even under non-strategic behavior, is an open research question [125, 122, 6, 14, 106].)

1.6 Thesis and Contributions

The thesis supported by my dissertation is that *a protocol based on bilateral barter between adjacent ISPs that act in their own interest can lead to efficient routing in the Internet and can be practically implemented.* By efficient, I mean that when ISPs use comparable metrics the routing quality is close to optimal routing which is globally optimized with complete information. By practical, I mean that the protocol preserves ISP autonomy, as per the requirements specified in Section 1.2, and its complexity, measured in terms of implementation, message, and processing requirements, is comparable to that of today's interdomain routing protocol.

The key contributions of my work are:

A novel approach for Internet routing with competing interests Wiser is based on a novel approach that combines bilateral barter and agnostic costs. Bilateral barter takes a holistic view of traffic exchanged between two adjacent ISPs, which enables efficient and win-win routing. Agnostic costs, or cardinal preferences, enable ISPs with diverse objectives to coordinate and limit the amount of information that ISPs are required to disclose. My evaluation shows that the combination of the two produces routing that is almost as efficient as potentially more complicated approaches based on multilateral coordination or global currency. It also suggests that simplifying my approach further would reduce efficiency.

A practical and efficient Internet routing protocol To my knowledge, Wiser is the first interdomain routing protocol that is both practical and leads to efficient routing. Wiser preserves ISP autonomy and has low overhead. It can be deployed in a framework that is similar to the current routing protocol. It retains today's simple monetary exchange practices in which payments between ISPs are coarsely tied to the amount of traffic exchanged, independent of the destination of the traffic. It also retains the current pair-wise contractual structure in which only neighboring ISPs have contracts with each other. Wiser is incrementally deployable in that two adjacent ISPs can use it to improve routing between them without waiting for deployment by other ISPs.

Understanding the impact of anarchy and autonomy in the Internet My evaluation provides insight into the impact of anarchy and autonomy in the Internet. I show that while the efficiency of anarchy in the Internet today is acceptable on average, likely due to network engineering by ISPs, it is poor for a small fraction of paths. The unreliability and operational cost associated with the manual control that is required to fix the poor paths suggests that the price of anarchy is high in the Internet. I also show that, for the topologies and workloads that I study, the efficiency of Wiser is uniformly high, which suggests that the price of autonomy is low.

To further understand the results above, I use simple analytic models to compute the efficiency of Wiser as a function of ISPs' internal costs of carrying traffic. The analysis predicts that Wiser is efficient when ISP costs are similar but inefficient otherwise. With dissimilar ISP costs, the efficiency is low because of the win-win requirement. That the efficiency of Wiser is high in practice suggests that the costs of ISPs that interconnect in multiple places tend to be similar.

1.7 Organization

The remaining chapters of this dissertation are organized as follows. In Chapter 2, I provide a brief background on Internet routing. I discuss the requirements for routing protocols for competing ISPs in Chapter 3 and describe Wiser in Chapter 4. In Chapters 5 and 6, I empirically and analytically evaluate Wiser. I review related work in Chapter 7 and conclude in Chapter 8.

Chapter 2

BACKGROUND AND MOTIVATION

In this chapter, I provide a background on Internet routing, with a focus on aspects that are relevant to this thesis, and illustrate the shortcomings of the current routing protocol.

2.1 Background

This tutorial is divided into two parts. The first part describes the current structure of the Internet, and the second part describes routing in the Internet.

2.1.1 Structural Organization of the Internet

From the perspective of routing, the Internet is an internetwork, or a network of networks. The individual networks are referred to as autonomous systems (ASes) and identified by their AS number which is a unique integer that serves as their identity. While some ASes are owned by organizations such as universities and corporations, others are owned by ISPs (Internet service providers), or commercial entities in the business of selling access to the Internet. The distinction between an ISP and an AS can be ignored for the purposes of this dissertation, and I use these terms interchangeably. Pairs of ASes connect to each other at one or more interconnections points. It is common for a large AS to interconnect to many other ASes and to have multiple interconnections, for example, in different cities, to other large ASes. An example internetwork is shown in Figure 2.1 in which each cloud represents a different ISP. In reality, ISPs have overlapping geographic coverage, but I show non-overlapping clouds for visual clarity. I also abstract away the network structure internal to the ISPs.

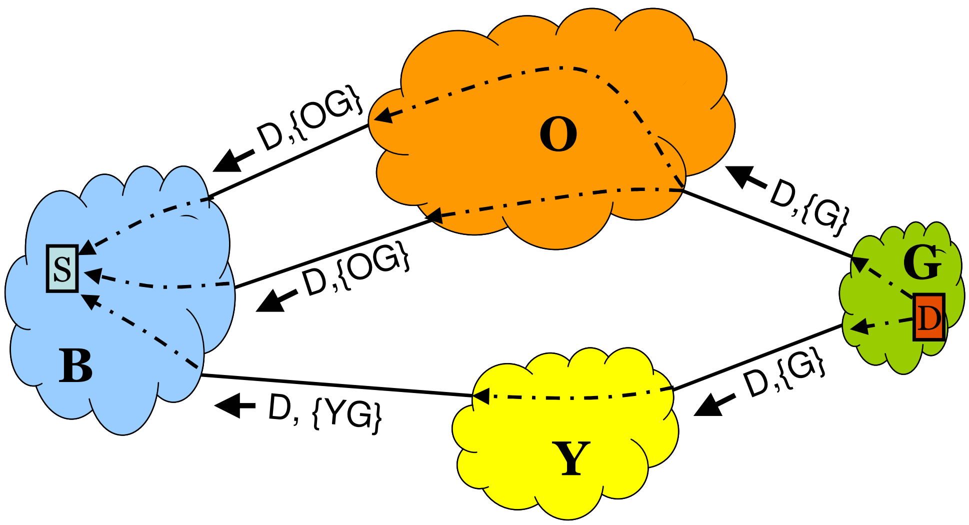


Figure 2.1: An example internetwork to illustrate the properties of Internet routing. Each cloud represents a different ISP and the solid lines between them represent interconnections. Routing information includes destination IP prefixes and AS-path and flows from the downstream to upstream ISPs, which is the opposite of the direction of data traffic.

Neighboring ISPs exchange traffic with each other based on bilateral contracts. End-to-end connectivity is achieved through a series of bilateral contracts. In Figure 2.1, traffic from B to G may go though O because of a contract between B and O and another contract between O and G. Contracts define the commercial relationship between the ISPs and the basis of monetary exchange, which tends to be simple in the current Internet. Common relationships include *customer-provider*, *peers*, and *siblings* [85, 45]. In a customer-provider relationship, the customer ISP pays the provider ISP to send as well as receive traffic. The amount of payment is usually a function of 95th percentile usage averaged over five minute intervals, i.e., maximum usage after discounting 5% of the peak usage intervals. Peers are often competitors that benefit from direct access to each other's customers. Typical contracts between peers require no monetary exchange as long as the ratio of the traffic in the two directions is within a factor of two. Siblings are friendly or related networks that provide unconditional and free transit to each other. Because the quantity of interest in current contracts is the total traffic crossing the ISP boundary, independent of the source or the destination of the traffic, ISPs are incented to minimize the resources a packet consumes inside their networks. This potentially enables them to carry more traffic.

Based on ISP relationships, the ISPs in the Internet can be informally classified into *tiers* [85, 116]. Tier-1 ISPs are the few biggest providers in the Internet and peer with each other. Tier-k ISPs are usually customers of ISPs that are tier-(k-1) or lower.

2.1.2 Routing in the Internet

As mentioned earlier, routing in the Internet operates at two logically distinct levels. *Intradomain* routing determines how ISPs route to destinations within their networks, and protocols for intradomain routing are known as interior gateway protocols (IGP). *Interdomain* routing determines how ISPs route to destinations outside of their network. This thesis focuses on interdomain routing because it is where ISP competition plays out.

For interdomain routing, ISPs use the Border Gateway Protocol (BGP) to exchange reachability information with each other [95]. As illustrated in Figure 2.1, routing information flows in the direction opposite to the flow of data, from downstream ISPs to upstream ISPs. The routing information today includes the IP address prefix specifying the destination and the list of ISPs along the path, known as the AS-path, to the destination. It does not include information on the performance or cost of the path. ISPs are reluctant to share this information because they worry that any disclosed information might be abused by their competitors. For instance, if an ISP discloses the monetary cost of carrying traffic between pairs of cities, a competitor can potentially infer the ISP's profitable paths and use that information to undercut the ISP's profits by adding more capacity of its own along those paths.

When multiple paths to a destination are available to an upstream ISP, it uses local policies to select the path. The commercial relationships with neighboring ISPs play a central role in these policies. Driven by monetary implications, ISPs usually prefer to send traffic through customers, peers and providers, in that order. Within these groups, ISPs usually select paths based on AS-path length, assuming that it reflects end-to-end path quality. Among paths with the same AS-path length, ISPs use local optimization criterion (see below) as the basis for path selection. Early-exit routing, in which ISPs select the locally optimal interconnection while sending traffic to their neighbors, is an example of such a path selection policy. The use of AS-path length as a basis for path selection suggests that, in addition to local concerns, end-to-end path quality (which customers value) is also important to ISPs. However, AS-path length is a poor indicator of end-to-end path quality in practice [118], and it is also often non-discriminating because many paths have the same length [90]

The local optimization criterion varies across ISPs. Being independent businesses, ISPs have different perceptions of what their customers want. Combined with the fact that the structure of different ISP networks is different [110], this leads to different optimization metrics. For instance, some ISPs minimize the average distance traversed by the traffic they carry, some ISPs minimize congestion, and yet others maximize the amount of traffic they can carry.

The original design of BGP [67] lacked support for optimizing network traffic, for instance, to balance network load, improve traffic performance, or reduce resource consumption. Over time, many ad hoc mechanisms have been added. Two mechanisms that are commonly used today are multi-exit discriminators (MEDs) and AS-path prepending. MEDs are used between two ISPs that interconnect in multiple locations to influence how traffic enters the downstream ISP. The downstream ISP attaches ordinal preferences to routing messages which encode how it wishes to receive traffic to that destination. Whether the upstream honors MEDs is contractually determined by the two ISPs. When honoring them, the upstream ISP selects the interconnection that is most preferred by the downstream ISP. With AS-path prepending, the downstream ISP artificially increases the AS-path length in routing messages going out via certain interconnections by adding its AS number multiple times. This reduces the traffic on that interconnection if the upstream ISPs consider this path to be longer and are less likely to use it. None of the current path selection mecha-

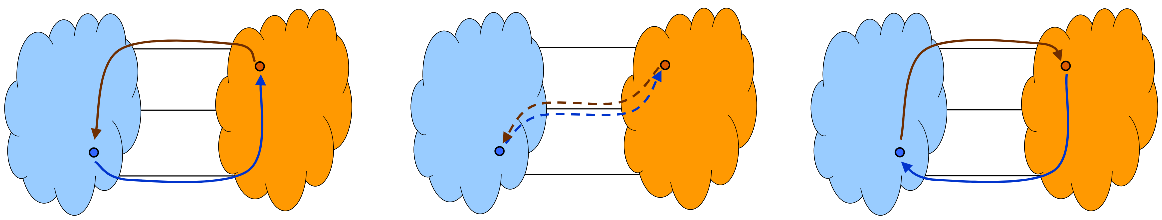


Figure 2.2: Early-exit routing leads to longer paths and a higher resource consumption inside both ISPs. (a) Early-exit routing pattern. (b) A routing pattern that is better for both ISPs. (c) Late-exit routing pattern that would emerge if MEDs are used.

nisms enable ISPs to jointly select paths. Either the upstream or the downstream is able to unilaterally determine routing paths.

2.2 Ill-effects of Internet Routing

ISPs in the Internet tend to prefer locally optimal paths without regard for other ISPs because BGP does not enable path selection that accounts for the individual concerns of all ISPs. The result is inefficient and unstable routing, which ultimately also increases the operational cost of ISP networks. Below, I use examples to illustrate these problems.

The first example concerns the efficiency of routing paths and the resource consumption inside ISPs' networks. Consider the two ISPs shown in Figure 2.2, with traffic flowing in both directions. Today, using early-exit routing, each ISP uses the locally optimal interconnection to transfer traffic to the other, resulting in the routing pattern of Figure 2.2(a). But consider the routing pattern of Figure 2.2(b) in which both ISPs use the middle interconnection. This pattern not only consumes fewer resources inside each ISP, which is the motivation behind early-exit routing, but also leads to shorter paths. Thus, locally optimal path selection hurts ISPs as well as applications. (Under certain topological assumptions the length of paths due to early exit routing can be up to three times that of shortest-path routing [54], though I show that it is usually less in practice.)

There is no straightforward way to achieve the routing pattern of Figure 2.2(b) with BGP. For instance, the use of MEDs leads to "late-exit" routing shown in Figure 2.2(c). When the ISPs agree to honor each other's preferences for incoming traffic, the traffic will use the link that is closest to the destination. Done consistently, this situation is simply the reverse of early-exit. For this reason, some ISPs manually look at the neighboring ISP's topology to decide if and for which destinations using MEDs might be useful [76].

While the situation above was hypothesized using the properties of Internet routing, such routing inefficiencies have been observed in practice. One such incident involved AT&T and Sprint, two tier-1 ISPs [109]. In this incident, traffic from San Francisco inside AT&T to San Francisco inside Sprint was being transferred at the Seattle interconnection even though the ISPs also interconnect in San Francisco. This was a result of AT&T optimizing its own network to avoid certain overloaded links, without any knowledge of the Sprint's network. Both ISPs would have been better off by coordinating their routing, which would have reduced resource consumption and improved traffic performance. The necessary coordination, however, could not have been achieved using currently available mechanisms.

My next example concerns managing overload after unexpected changes in the topology or traffic such as failures or flash crowds. Consider the two ISPs in Figure 2.3(a), with traffic flowing from ISP-A to ISP-B. Assume that the middle interconnection fails and ISP-A reroutes the affected traffic based on local conditions, as shown in Figure 2.3(b). Suppose this overloads ISP-B, which reacts by shifting some traffic to the top interconnection, as shown in Figure 2.3(c). Now suppose that the action of ISP-B overloads ISP-A, and it reacts by shifting traffic back to the bottom link, as shown in Figure 2.3(d). The result is a return to the situation of Figure 2.3(b), and continuing the cycle of influence. Figure 2.3(e) shows a routing pattern that is acceptable to both ISPs. But, as before, there is no straightforward way today to discover this configuration.

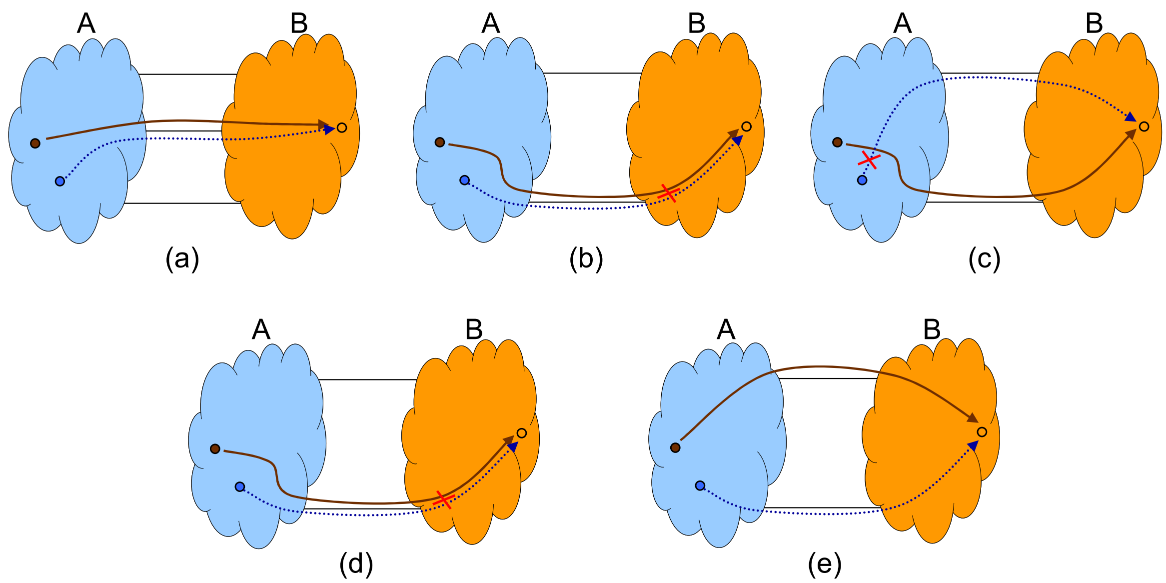


Figure 2.3: Unilateral traffic movements lead to instability. (a) The routing pattern before the failure of the middle interconnection. (b) ISP-A responds to the failure by moving both flows to the bottom interconnection, which congests ISP-B. (c) ISP-B reacts by moving the dotted flow to the top interconnection, which congests ISP-A. (d) ISP-A reacts by moving the dotted flow back to the bottom interconnection, which again congests ISP-B. (e) A routing pattern that is acceptable to both ISPs.

This example is adapted from a real incident between two large ISPs. Typically, ISPs do not react automatically to overload in their networks; in the incident, operators were manually moving the traffic by changing their path selection policies and the routing messages they sent to each other. They were stuck in this cycle for two days before they discovered the root cause of the problem and settled on a mutually acceptable solution.

The problem above can be considered another form of inefficiency – in terms of network provisioning – with the current routing. To avoid such instabilities, ISPs highly overprovision their networks today. A better routing protocol can reduce the provisioning required by making ISPs respect other ISPs' capacity constraints. (While most ISPs overprovision today, the provisioning level was insufficient to avoid the problem in this case.)

A common theme in the examples above is that all ISPs suffer because of locally optimal path selection. In this sense these problems are not a fundamental consequence of competing interests, and they can be addressed if ISPs have a protocol to facilitate cooperation. To be practical, such a protocol must reflect the interests of ISPs. For instance, in the first example, sharing more information by itself is not sufficient to achieve the routing pattern of Figure 2.2(c). The left ISP may not move its traffic, even when it knows that the movement leads to shorter end-to-end paths, unless the right ISP moves its traffic as well. Otherwise, the resource consumption inside the left ISP would be higher compared to doing nothing. The protocol must ensure that neither ISP loses. In the next chapter, I discuss this and other requirements that a practical routing protocol must meet.

Chapter 3

PROBLEM AND SOLUTION REQUIREMENTS

In this chapter, I state in detail the problem that my work addresses and the requirements that any solution must satisfy to be practical.

3.1 Problem Statement

Consider an internetwork of many ISPs. Each ISP is a network of nodes and links. While ISPs have complete information about their networks, such as latency and utilization level of internal links, they know little about other ISPs' networks. The traffic in this internetwork consists of a set of flows, where a flow is the collection of packets between two distinct nodes. The path of a flow is the sequence of nodes traversed by it, possibly across multiple ISPs. The routing pattern is the set of paths for all the flows.

ISPs incur a cost for each flow that traverses their network. Different ISPs measure this cost differently, based on the internal state of their network, their optimization goals, and the path of the flow. I assume that each ISP aims to minimize its total cost of carrying traffic, which is the sum of the costs of individual flows that traverse its network.

My goal is to design a *practical* interdomain routing protocol that enables ISPs to compute *efficient* routing patterns. Recall that an interdomain routing protocol helps ISPs determine paths to destinations outside of their network. Properties that make a protocol practical in this context are specified in the next section. The ideal efficiency goal for a routing protocol is social optimality, i.e., it should minimize the sum of costs incurred by all ISPs. But when ISPs use incomparable cost measures, socially optimal routing is not defined. My goal is that the efficiency be close to a Pareto-optimal routing pattern. A routing pattern is Pareto-optimal if all other routing patterns have a higher cost for at least one

ISP. Pareto-optimality rules out routing patterns with obvious wastage, i.e., those that have a higher cost for all ISPs. The current Internet routing, for instance, is often not Pareto-optimal because a different routing pattern can lead to a lower cost for everyone. There can be many Pareto-optimal solutions in a system. I want to approximate one of the solutions that come close to being socially optimal when ISPs use comparable measures, under the constraint that the cost of each ISP is no more than what it incurs today.

3.2 Requirements for a Practical Protocol

For a routing protocol to be deemed practical in the Internet context, it must fulfill two sets of requirements. The first set includes traditionally valued properties such as scalability, stability, low implementation complexity, low routing message overhead, and fast convergence. The second set of requirements stem from the competitive nature of ISPs and are specifically relevant to this thesis. Below, I identify four such requirements, mention the aspects of my approach that are geared towards them, and point out why some alternative approaches are not suitable.

Limited information disclosure Due to competitive concerns, ISPs are reluctant to disclose sensitive internal information. ISPs usually consider detailed information on the topology and performance of their networks as being sensitive. This sensitivity also extends to monetary cost, since an ISP may not wish to tell its competitor the monetary cost of carrying traffic.

I handle this concern by having ISPs disclose *agnostic* costs rather than requiring them to disclose transparent metrics such as latency or monetary cost. Agnostic costs are cardinal preferences of an ISP that are derived from its internal optimization criterion. ISPs do not disclose their optimization criterion or the derivation methodology. The extent of information revealed through agnostic costs is more than what ISPs are required to disclose today through MEDs because while MEDs specify ordinal preferences, agnostic costs are cardinal. For the topologies that I study in Chapter 6, I show that ordinal preferences do not lead to efficient routing.

An alternative to limiting the amount of information disclosed as part of routing is to employ incentive compatible mechanism design in which the ISPs reveal their internal information such as monetary cost [38]. In this approach, concerns regarding information disclosure are addressed by formally proving that honest disclosure is the best strategy for any ISP even if it has complete information about the other ISPs. This implies that an ISP cannot unfairly gain using this information. However, information disclosed as part of routing can be abused outside of the routing framework [40]. For instance, if a competitor knows about an ISP's profitable routes, it can add capacity along those paths to undercut the ISP's profits. This makes (direct) mechanism design inappropriate for our target environment.

Support for heterogeneous objectives Internet routing protocols must enable ISPs with different optimization goals to cooperate. ISPs in the Internet have different optimization criteria, depending on their networks. For instance, while ISPs with capacity constraints may aim to avoid overload, those with overprovisioned networks may aim to improve performance by reducing latency and jitter. Yet others may want the best routes for their preferred customers. There are certainly other considerations of which I cannot be aware. Since there is no universal optimization metric, instead of focusing on one or more chosen metrics, routing protocols should be agnostic toward the metric used by individual ISPs.

Agnostic costs already fulfill this requirement. An alternative approach is to use monetary cost as a unifying metric. In addition to information disclosure, a problem with using monetary cost is that it can be very difficult, if not impossible, for ISPs to quantify their internal considerations in terms of monetary cost [107]. In contrast, it is easy for ISPs to map their internal objectives to agnostic costs; many ISPs already perform a similar mapping as part of optimizing intradomain traffic.

However, routing protocols that support diverse objectives without the use of monetary cost cannot meet social goals such as social optimality or fairness. Social goals are not well-defined when entities have incomparable objectives. For instance, social optimality is undefined when one ISP minimizes latency and the other minimizes link utilization. This is why my goal is a Pareto-optimal routing pattern.

Individually beneficial In an environment with selfish, profit-maximizing parties, it is imperative that, compared to unilateral behavior, individual entities not lose by cooperating. I refer to this property as *win-win*, and in its absence, ISPs that stand to lose will not be inclined to adopt the protocol. The loss and gain is measured differently by each ISP, but in general it will be a combination of the ISP's local objectives and the end-to-end performance experienced by its customers. While all efficient routing protocols are expected to result in social gain, not all of them are guaranteed to result in individual gain. Since routing paths today are locally optimal for at least one of the ISPs, changing an individual path may represent a loss for that ISP. A protocol that does not consider the gain or loss to individual ISPs might result in routing that causes some ISPs to lose.

Based on the observation that the interaction between ISPs is not limited to individual flows but spans across multiple flows and across time, my approach is to enable ISPs to barter. ISPs trade favors in a manner that results in win-win outcomes.

An alternative to barter is to use monetary compensation. ISPs that win pay the losing ISPs such that the overall result is win-win. However, the use of real money runs into the problems mentioned above and also raises the barrier for adoption by going against current practices of using simple monetary exchange criteria. For instance, most peering relationships involve no money transfer despite the potential for one ISP to benefit more than the other. A potential advantage of money over barter is that it may lead to more efficient routing. However, I find that barter does equally well for the realistic ISP topologies and workloads that I study.

Robustness to cheating A concern when competing entities interact is that one of them may cheat, that is, violate the protocol to manipulate the outcome in its favor. The protocol must discourage such manipulation. Incentive compatible mechanisms, in which truth telling is provably the best strategy for all entities, guarantee interactions that are provably strategy-proof. However, provable incentive compatibility often runs counter to efficiency. It is known that in the absence of a third party acting as a subsidizer, appraiser or arbitrator, there does not exist a budget-balanced mechanism that is both incentive compatible and able to implement all mutually acceptable solutions for bilateral trading [82, 21]. This result is applicable to ISP routing, and because a budget-balanced mechanism is desired and there are no natural third parties, I face a trade-off between incentive compatibility and efficiency.

In this trade-off, I favor efficiency so that efficient solutions can be computed for the common case of honest ISPs. (And even if I were to favor incentive compatibility, a mechanism design approach fails to fulfill the other requirements above.) I believe that honesty will be the common case, because in general, competing parties tend to act honestly while seeking joint gains over a default contract [93]. For instance, even today ISPs often cooperate using ad hoc mechanisms that are not inherently robust against cheating.

Specifically, cheating is a poor long-term strategy for ISPs for three reasons. First, a good reputation is invaluable from a business perspective, and getting caught has serious consequences, such as monetary penalties or disconnection from other ISPs. Even in the absence of specific detection mechanisms, usually there is enough ancillary information in the system to detect persistent cheating, especially if it is egregious (e.g., see reference [78] from the phone network). Second, ISPs coordinate because it improves their internal routing, and an ISP may stop coordinating if it experiences limited gain due to cheating by a neighboring ISP. As a result, the cheating ISP might be worse off than being honest. Third, the indirect cost or effort required to cheat effectively can dwarf potential gain. Effective cheating strategies are those that lead to worthwhile gain for the cheater, keep the neighboring ISP interested in playing, and do not lead to much performance degradation for traffic so that the ISP's own customers do not suffer. Even if such strategies exist, they will likely require detailed information about other ISPs' networks and traffic. Collecting such information is difficult.

However, favoring efficiency over incentive compatibility does not imply that I want to allow a cheating ISP to infinitely game the system. In fact, it is essential that ISPs that are tempted to cheat, even if only in the short-term, experience limited gain. This removes the incentive to continue cheating. It is also desirable that the loss to honest ISPs be small. My approach is to restrict the degree of freedom available to ISPs to limit the gain for cheating ISPs and the loss for honest ISPs.

Chapter 4

DESIGN AND IMPLEMENTATION OF WISER

I now describe the design and implementation of Wiser. I start in Section 4.1 by outlining my approach. In Sections 4.2 and 4.3, I describe the design of Wiser, first for the case of two adjacent ISPs and then for the case of multiple ISPs. In Sections 4.4 and 4.5, I discuss how ISPs might assign agnostic costs to their paths and the robustness of Wiser to cheating. I conclude this chapter in Section 4.6 where I describe how Wiser can be implemented in the Internet within the framework of current interdomain routing.

4.1 Approach

In the last chapter, I motivated barter and agnostic costs in the context of requirements that stem from the need to preserve ISP autonomy. There are two methods to barter in a network with multiple ISPs. The first method is bilateral barter involving pairs of ISPs. In this, ISPs aim to gain in their interaction with each ISP with which they barter. The second method is multilateral barter that involves groups of more than two ISPs. The entire internetwork is a potential barter group, and so are smaller groups such as those involving ISPs that operate in a given geographic region. In multilateral barter, ISPs barter with the rest of the group with the aim of gaining in the aggregate, that is, each ISP trades with the rest of the group without distinguishing between different ISPs within the group.

I use bilateral barter between adjacent ISPs because it is simpler to implement and enforce. It does not require global, multi-party coordination and closely mirrors the contractual structure of the Internet. It enables ISPs to treat their neighbors differently, which is difficult to accomplish with multilateral barter. It can also makes collusion by two or more ISPs against another ISP less effective because ISPs have independent relationships with each neighbor. However, multilateral barter is likely to be more efficient just as bigger markets are more efficient. ISPs will be willing to lose a little in their interaction with one neighbor if they gain more from another, leading to a higher overall gain. I show that bilateral barter is equally efficient for the topologies and workloads that I consider in Section 5.

I argue that ISPs will be willing to disclose agnostic costs to other ISPs. With agnostic costs, ISPs do not disclose transparent metrics, such as distance or monetary cost, but disclose a mapped version of the (unknown to others) internal criterion. If the internal criterion is not strictly performance-based, it will be hard for others to make sense of the agnostic costs. But if the internal criterion is performance-based, such as distance, it is possible that other ISPs can use the agnostic costs to infer performance. However, this is hardly a new capability; even today, ISPs can measure performance inside each other's network. Additionally, by not requiring ISPs to use a specific criterion or derivation methodology, agnostic costs enable them to control the trade-off between the efficiency of routing and the amount of information disclosed. For instance, coarsely mapping internal criteria to agnostic costs discloses less information but may lead to less efficient routing; precise mappings disclose more information but are likely to lead to more efficient routing.

Figure 4.1 illustrates my approach. This example is adapted from Figure 2.2, and I use it as the canonical example in this chapter. Assume that the left ISP optimizes for the number of hops and the right ISP for the length of the path. Figure 4.1(a) shows the internal cost of carrying traffic between the nodes shown inside their network and the interconnection. For ease of exposition, assume that these costs are symmetric, i.e, the cost of the path between two nodes inside an ISP is the same as the cost of the reverse path. (Wiser does not make this assumption.) Figure 4.1(b) shows the routing that will be produced today as each ISP picks the locally optimal interconnection. The numbers represent agnostic costs, assuming that the ISPs use the identity function to map their internal metrics to agnostic costs. Of course, this routing pattern is sub-optimal with costs of 8 (=1+7) to the left ISP and 12 (=1+11) to the right ISP. Figure 4.1(c) shows that using the middle interconnection is better for both ISPs, with respective costs of 4 (=2+2) and 6 (=3+3). If ISPs consider only one

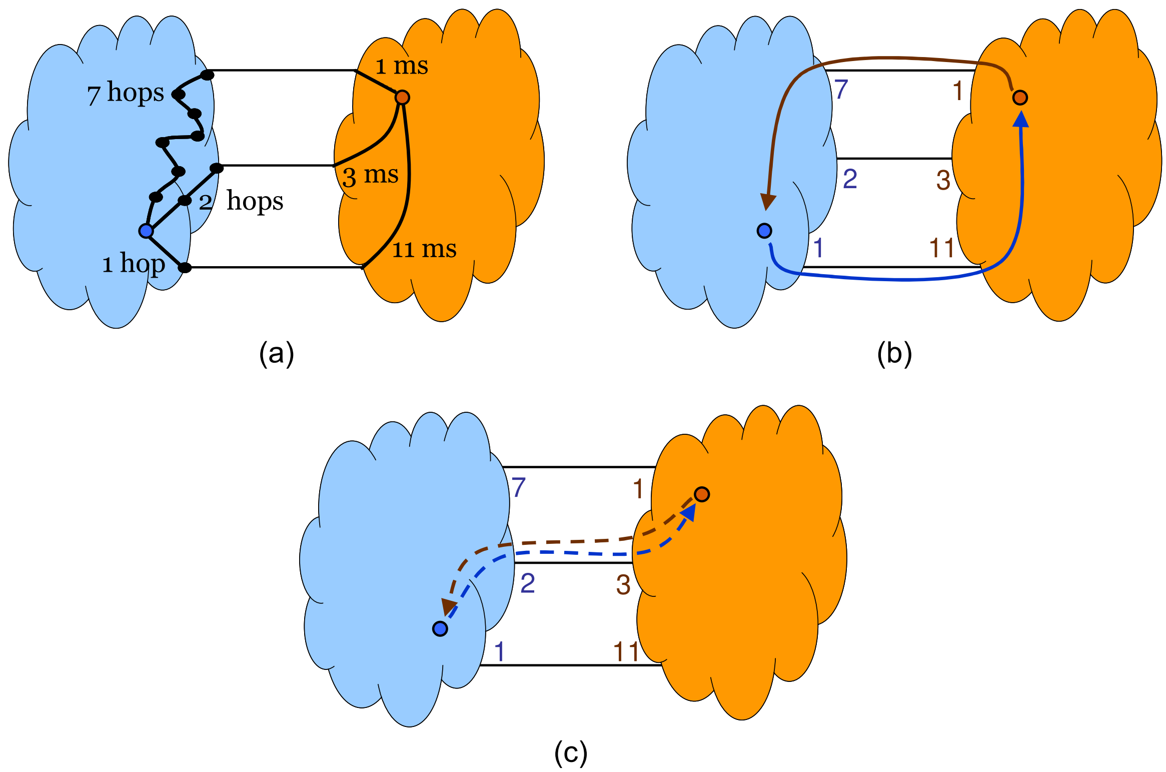


Figure 4.1: An example of barter with agnostic costs. (a) The internal optimization criteria of the two ISPs. (b) The agnostic costs and the routing pattern with locally optimal path selection. (c) The routing pattern with barter: each ISP compromises a little on one flow for greater gains on the other flow, such that the overall gain is positive.

flow at a time, they have no incentive to move their flows. But by considering both flows together, a barter can be arranged such that each ISP is willing to move its flow in return for the other doing the same.

Observe that barter with agnostic costs does not explicitly optimize end-to-end paths. For instance, the barter above, where one ISP optimizes hop count and the other optimizes length, does not necessarily minimize either the hop count or the length of end-to-end paths. Instead it implicitly improves end-to-end paths by improving the situation for each ISP. In practice, this can be sufficient to avoid egregiously bad inter-ISP routing paths, bringing

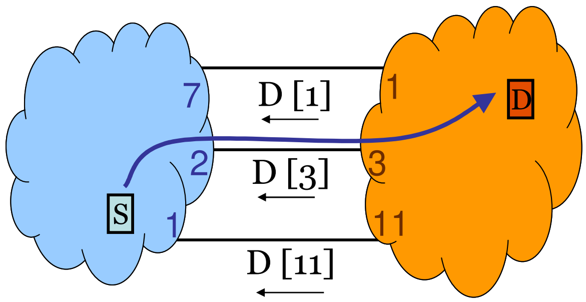


Figure 4.2: Wiser in the two ISP case. The numbers inside the upstream ISP (on the left) represent its agnostic costs of carrying traffic between S and the corresponding interconnection, and those inside the downstream ISP represent its agnostic costs of carrying traffic between the interconnections and D. With Wiser, the downstream ISP discloses its agnostic costs to the upstream ISP, and the upstream ISP selects an interconnection based on both its own and the downstream ISP's costs.

most of the benefit of a more direct but possibly more complicated optimization scheme. Researchers have observed in other systems that avoiding egregiously bad cases brings the most gain. For instance, when using multiple servers in a content distribution network, the response time achieved by simply avoiding distant servers is similar to that achieved by consistently using the optimal server [55].

4.2 Two-ISP Case

I now describe the design of Wiser. For ease of exposition, I first describe the design in the context of two ISPs and then extend it to multiple ISPs in the next section.

In the two-ISP case, Wiser enables ISPs to judiciously select interconnections for the traffic they exchange. Consider traffic going from S to D in Figure 4.2. The figure also shows the hypothesized agnostic costs of the two ISPs. Wiser operates in a framework that is similar to shortest-path routing. For each destination, the downstream ISP advertises to the upstream ISP its agnostic costs of carrying traffic from each interconnection to the destination. The upstream ISP selects an interconnection based on *both* the local costs of carrying traffic from the source to various interconnections and the agnostic costs advertised by the downstream ISP. One possible way to accomplish this is by selecting paths that minimize the sum of the two costs; the real constraint on path selection will become clear below. The key is that the upstream ISP compromises by not choosing locally optimal paths. Barter happens because each ISP is upstream for some traffic and downstream for some. ISPs compromise when they are upstream in order to benefit when they are downstream.

While the procedure above can produce efficient routing, it can be easily gamed. Downstream ISPs can lie about their agnostic costs, and upstream ISPs can select locally optimal paths. I now describe mechanisms to address these problems. As discussed previously, these mechanisms are not designed to be perfectly strategy-proof but to limit the gain that is possible through cheating.

4.2.1 Dishonest cost disclosure

If the downstream ISP can advertise arbitrary agnostic costs, and it chooses to be dishonest about its costs, the upstream ISP will find it difficult to select paths in a way that simultaneously preserves its own interests and respects the downstream ISP's costs. Figure 4.3 illustrates this point. The routing that emerges when the downstream ISP is honest is shown in Figure 4.3(a). But the downstream ISP can be dishonest about its costs. For instance, as shown in Figure 4.3(b), it can inflate them by a factor of ten. If the upstream ISP continues to select path that minimize the sum of costs, the resulting routing pattern has a higher cost for it than the original pattern.

To incent honest cost disclosure, upstream ISPs in Wiser normalize downstream ISP's costs, guided by the principle that both ISPs are equal partners in this barter. One way to normalize the costs is for the sum of the agnostic costs announced by both ISPs to be the same, where the sum is computed across all destinations announced via all interconnections. Normalization is done by multiplying the incoming costs by the normalization factor, i.e., the ratio of the unnormalized sums. Figure 4.3(c) shows the routing pattern

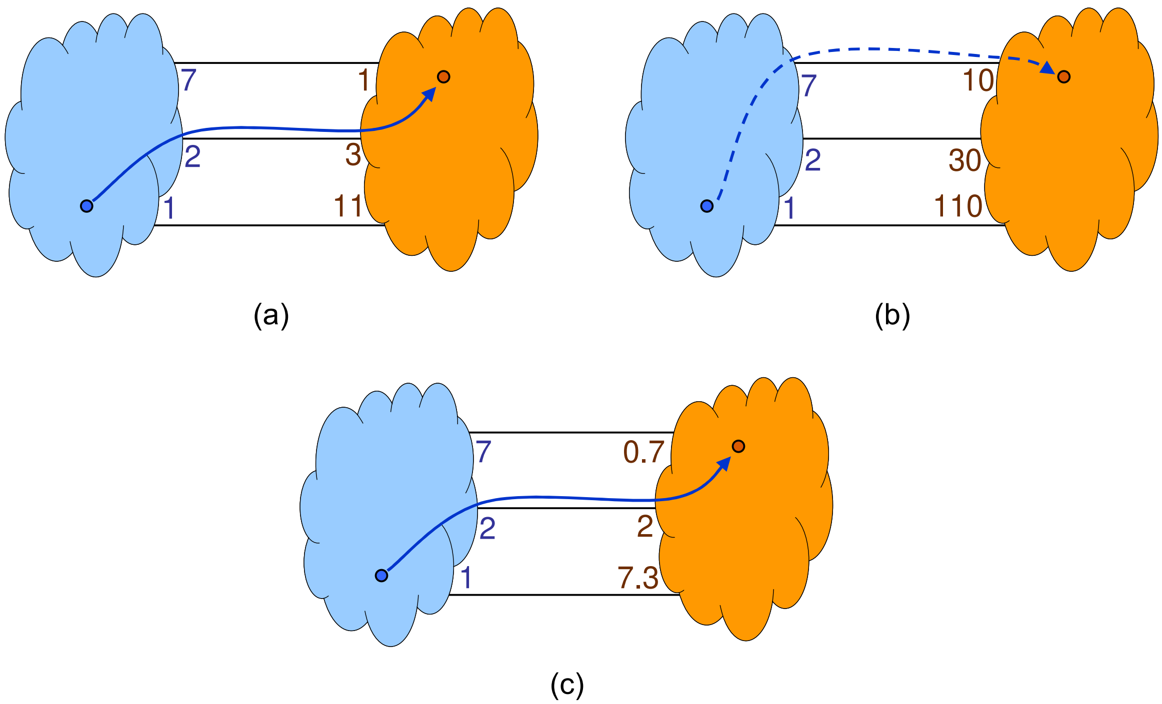


Figure 4.3: Incenting honest cost disclosure. (a) The real agnostic costs for the downstream ISP and the resulting routing pattern. (b) The routing pattern that emerges if the downstream ISP inflates its costs by a factor of ten. (c) Normalized downstream ISP's costs and the resulting routing pattern.

that emerges after the downstream ISP's costs have been normalized such that they sum to ten, which is the sum of the upstream ISP's costs. Computing the normalization factor requires information sharing among routers of an ISP that connect to another ISP. I explain in Section 4.6 how this can be accomplished using already deployed mechanisms.

ISPs can also use other normalization schemes if they want to explicitly account for any significant asymmetry between them, such as the number of destinations they announce to each other or the amount of traffic they send to each other. For instance, a higher weight can be assigned to the costs of the ISP that sends more traffic. Similarly, provider ISPs can choose to assign a higher weight to the costs of their customer ISPs. My experiments, however, use the normalization mechanism described in the previous paragraph.

Normalization limits the gain that an ISP can achieve by lying about its agnostic costs. Uniformly inflating all costs, as in Figure 4.3(b), does not impact routing. With normalization, selectively increasing some costs automatically decrease the remaining ones. This might cause the upstream ISP to start using paths that appear cheap (to the upstream ISP) but are actually costlier for the downstream ISP. For instance, in Figure 4.3(c), if the downstream increases the cost of the middle link, it effectively decreases the cost of the other two links. When multiple sources inside the upstream ISP are sending traffic to the downstream ISP, some of them might start using links that are costlier for the downstream ISP. The downstream ISP might consider using detailed information on traffic and the upstream ISP's costs, if known, to try to extract some advantage. But normalization limits possible gain by reducing the available degrees of freedom. I empirically demonstrate this in Section 5.5 for a cheating strategy in which the downstream ISP has complete information about traffic and the upstream ISP's costs.

In addition to providing robustness towards dishonest cost disclosure, normalization facilitates win-win routing. By making the ISPs equal partners in the barter, it makes path selection sensitive to the concerns of both ISPs. For instance, consider a situation in which one ISP's agnostic costs are in the range [0-10] and the other ISP's agnostic costs are in the range [0-1000]. Here, vanilla lowest cost routing will choose paths that favor the second ISP because its agnostic costs will dominate, but normalization enables an equal-footing comparison across the two sets of agnostic costs. Using realistic topologies and workloads, I show in Chapter 5 that this leads to win-win routing most of the time. But when there is not much overall gain to be had through cooperation, some ISPs can lose a little relative to the routing today. Such ISP pairs can revert to the way they route currently without impacting the efficiency of routing paths.

Normalizing incoming agnostic costs also enables an upstream ISP to compare the agnostic costs received from different downstream ISPs. As used today, the MEDs re-

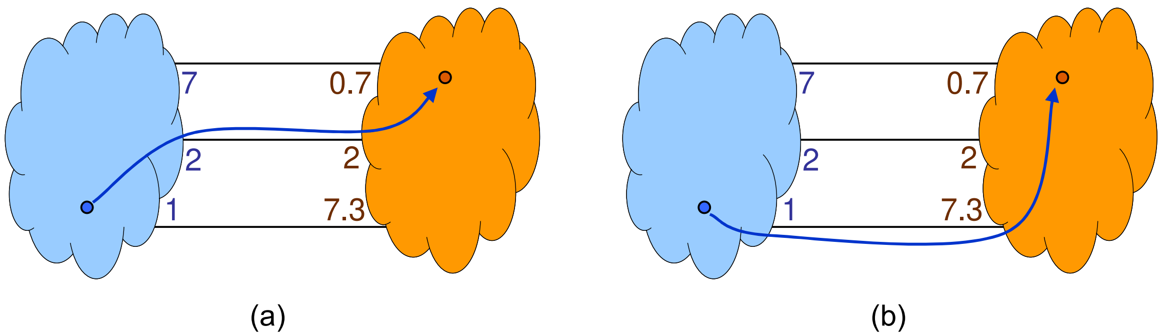


Figure 4.4: Dishonest path selection. The numbers represent the agnostic costs of the two ISPs, normalized to the same sum. (a) Routing pattern when the upstream ISPs is honest. (b) Routing pattern when the upstream ISP is dishonest.

ceived from different ISPs are incomparable, which creates practical problems for the recipient [114].

4.2.2 Dishonest path selection

I now describe how Wiser incents upstream ISPs to be sensitive to downstream ISPs' costs while selecting paths. In the absence of such an incentive, an upstream ISP might continue to select locally optimal paths, undermining the rationale for the downstream ISP to respect costs for the reverse paths. Figure 4.4 illustrates this point. On the left I show the normalized costs of the two ISPs and the routing pattern that emerges when both ISPs are honest. But (based on what I have described so far) nothing stops the upstream ISP from using the bottom interconnection, as shown in Figure 4.4(b), which has a higher cost for the downstream ISP. It is not straightforward for the downstream ISP to verify when that happens because in some instances the internal costs of the upstream ISP can be such that the bottom interconnection is the best overall path.

To incent honest path selection, Wiser uses a combination of virtual payments and a contractual clause. When an upstream ISP sends traffic to a destination over an interconnection, it makes a virtual payment to the downstream ISP. The amount of payment equals the number of bytes sent multiplied by the agnostic cost announced by the downstream ISP for that destination over that interconnection. Now consider how the average payment per byte across all flows sent by the upstream ISP differs with the path selection policy. If the upstream ISP is dishonest and is oblivious to the downstream ISP's costs, according to the law of large numbers, the average payment is expected to be roughly equal to the average cost announced by the downstream ISP across all destinations over all interconnections. With honest path selection, the average payment is expected to be less than the average announced cost because the upstream ISP avoids paths that are costly for the downstream ISP. Wiser leverages this expected behavior by having ISPs sign contracts stipulating a bound on the ratio of the average payment to the average announced cost. The averages can be measured using either unnormalized or normalized agnostic costs because the ratio is independent of normalization.

Let us revisit the example above, this time with virtual payments. The average announced cost from the downstream ISP to the upstream ISP is 3.3 ($\frac{0.7+2+7.3}{3}$). Assuming unit flow size, the average payment that the upstream ISP makes is 2 if it uses the middle interconnection and 7.3 if it uses the bottom one. Thus, the ratio of average payment to average cost is $\frac{2}{3.3}$ when it is honest and $\frac{7.3}{3.3}$ when it is dishonest. If the latter ratio is higher than the contractual bound, it will lead to monetary penalties, discouraging the upstream ISP from such behavior.

The contractual clause above is inspired by current contractual practices between ISPs. It is similar to the clause concerning bounds on traffic ratios mentioned in Section 2.1.1 and involves no real money transfer in the common case. It gives each ISP flexibility in path selection, while incenting them to prefer low cost paths. In essence, Wiser gains online robustness by leveraging offline contracts between ISPs, while keeping the design simple.

Contractual clauses other than the one specified above are also possible in this framework. For instance, a provider ISP can charge customer ISPs based on the value of the ratio to encourage them to send traffic along paths that are cheaper for the provider (even if that leads to higher internal costs for the customer).

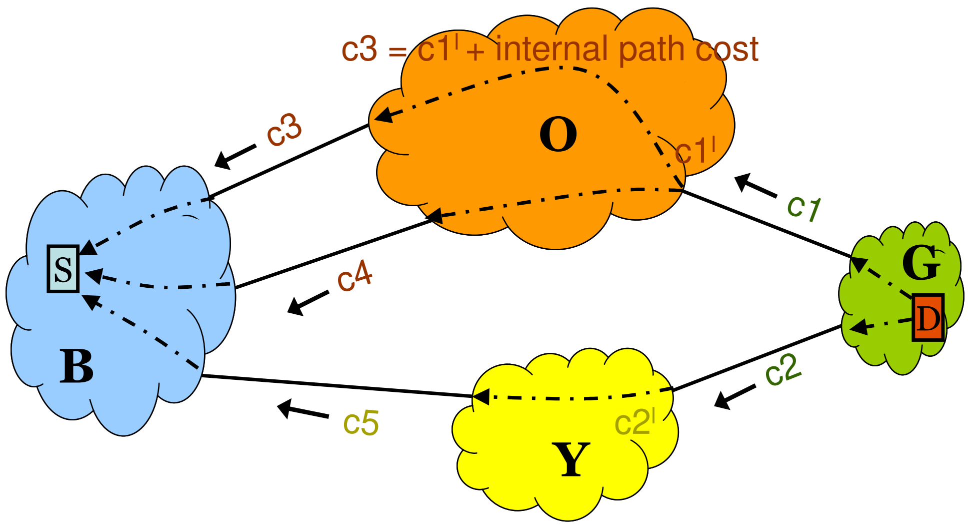


Figure 4.5: Wiser in the case of multiple ISPs. Downstream ISPs announce agnostic costs to upstream ISPs. ISPs normalize incoming agnostic costs and propagate them after adding their internal path costs. To select paths (not shown in the figure), the source considers both the local and received agnostic costs.

This completes my description of the design of Wiser for the case of two ISPs. I next describe how Wiser works in the general case of multiple ISPs.

4.3 Multi-ISP Case

Wiser works as follows for the case of multiple ISPs. Figure 4.5 illustrates my description.

- ISPs announce internal destinations to their neighboring ISPs. The announcements contain the agnostic cost of the path from the interconnection to the destination. The figure shows G announcing the destination D to O and Y with a cost of $c1$ and $c2$.

- ISPs normalize the costs received from their neighbors. The conversion is based on the normalization factor between the two ISPs, which as specified earlier, is computed based on all cost announcements flowing between the two ISPs. The figure shows $c1$ and $c2$ being converted to $c1'$ and $c2'$.

- When ISPs propagate a routing announcement that they received from another neighbor, they add their internal cost to the cost received from the neighbor. The figure shows O adding its internal cost to $c1'$. The addition enables bilateral barter to optimize over complete paths. The contract between O and B incents B to be sensitive to $c3$, and the contract between O and G encourages O to reflect $c1$ while computing $c3$. In each case, the incentive is to receive fair treatment for returning traffic.

- When an ISP receives multiple paths to the same destination, it selects a path based on local and received costs. In the figure, B will convert the incoming costs $c4$, $c5$ and $c6$, and it will then select an interconnection based on the converted costs and the costs of the paths from the source to each of the interconnections.

Section 4.6 describes in detail how the protocol above can be implemented within the current framework of Internet routing.

4.4 Deriving Agnostic Costs

Wiser places no restriction on how ISPs derive their agnostic costs except for the underlying assumption (in Section 3.1) that ISPs aim to minimize the sum of the costs of individual flows. For this, ISPs need to derive agnostic costs such that minimizing their sum leads to optimizing their objective. This is straightforward for objectives that are based on additive path properties such as propagation delay, hop count, or queuing delay. For instance, the identity function can be used to minimize the average propagation delay. For other objectives, such as minimizing maximum link utilization, mappings similar to those proposed by Fortz and Thorup can be used [42]. They use a piece-wise linear function with increasing slope to assign a cost to a link based on its load such that minimizing the sum of these costs approximately minimizes the maximum link utilization.

As an optimization to hide small changes in internal paths, ISPs can also use measures that change less frequently. For instance, they can use the geographic distance between the end-points of the internal path [84] to approximate propagation delay; unlike propagation delay, geographic distances hide changes in internal paths.

Stability Stability of routing based on load-dependent metrics has been a long-standing question of interest in large-scale networks [44, 59, 125, 122, 6, 14, 106]. A stable routing protocol is one that converges when the load on the network or the topology of the network no longer changes. While I leave investigating the algorithmic stability of Wiser to future work, I note that capacity overprovisioning in today's networks helps with stability. It makes link overload an infrequent occurrence, and shedding load on an overloaded link is easier when doing so does not overload other links.

I specify guidelines, extracted from previous research, for deriving agnostic costs in a way that enhances routing stability. For stability, it is necessary that the path costs change at a rate that is slower than the reaction time of the routing [14, 106]. This can be achieved by following two guidelines. First, the costs should not be closely coupled to load. For instance, they should change only after the load changes by 10% rather than 1%. A potential concern with coarse dependence is that the ISP cannot react to smaller changes in the link load, which might require more capacity if congestion-free operation is desired. However, if the network is well-provisioned to deal with a broad range of failures, reacting to small changes is not important; I show evidence of this in my evaluation. Second, costs should be based on slowly changing measures of link load. For instance, measures such as instantaneous queue size or link utilization computed on small time windows can change very rapidly but costs based on the amount of traffic in long-lived flows leads to stable routing [106]. Existing work on quality of service (QoS) routing provides other guidelines on setting such dynamic costs [34, 126, 8].

While the above guidelines cannot be enforced without impinging on ISP autonomy, there is competitive pressure on ISPs to ensure that their agnostic costs do not change rapidly. If an ISP's costs change too frequently, other ISPs are likely to ignore its routing announcements, as is done today with route flap damping [121]. Another possibility (which

I have not yet explored) is to penalize ISPs that change their costs faster than a contractually specified rate.

4.5 Robustness to Cheating

I consider cheating to be a behavior in which ISPs deviate from the protocol specification to either gain unfair advantage or hurt other ISPs. In Wiser, an ISP can cheat by either announcing costs that are not reflective of its true agnostic costs or by dishonestly selecting routing paths. Recall that even though cheating in Internet routing is not expected to be common, if an ISP decides to cheat, it is desirable that the gain for cheating ISPs and the loss for honest ISPs be small. In Section 5.5, I study several cheating strategies in which the cheater focuses on self-gain, and I show that the normalization and payment ratio constraints limit the gain for the cheater.

While I do not empirically evaluate scenarios where the cheater focuses on hurting a neighboring ISP, I argue that Wiser limits the loss to honest ISPs. For traffic going from the cheater to the victim, the cheater can accomplish this by selecting paths that are highly sub-optimal for the victim. However, the success of this strategy is limited by the virtual payment ratio because the cheater cannot select paths that increase the ratio beyond the threshold. For traffic going from the victim to the cheater, the cheater can hurt the upstream neighbor by modifying the announced costs such that the neighbor selects paths that are highly sub-optimal for itself. However, the success of this strategy is also limited because the upstream ISP always takes its own costs into account while selecting paths.

4.6 Internet Implementation

In this section, I describe an implementation of Wiser for the Internet. To lower the barrier for deployment, this implementation mirrors the framework in which BGP is implemented today. It is also incrementally deployable in that two neighboring ISPs can start using Wiser to improve routing for the traffic they exchange without waiting for deployment by other

ISPs. I have implemented Wiser on top of two independent platforms, SSFNet [113] and XORP [129], and I use these implementations to evaluate its overhead in Chapter 5.

My implementation assumes that the ISP uses a cost-based IGP (interior gateway protocol) such as OSPF [80] for intradomain routing, and it uses IGP costs as the agnostic costs. That is, the internal component of the agnostic cost to the destination is the IGP cost to the egress router within the ISP network. This is similar to how many ISPs today use these costs as the basis for MEDs [75, 76] (even though MEDs are only required to be ordinal). Straightforward extensions can accommodate ISPs that do not want to use IGP costs as agnostic costs or ISPs that use non-cost-based intradomain routing, for instance, routing based on MPLS [96].

In the discussion below, a *route* refers to a routing message passed between ISPs, which includes the destination and the associated routing information. Assuming that ISPs are already running an interdomain routing protocol similar to BGP, implementing Wiser entails the following.

- In addition to other attributes, routing messages between ISPs have agnostic costs attached to them. I define a new optional, non-transitive attribute for this purpose. A new type of community attribute [26] can also be used instead. As previously described, for routes originated by the advertising ISP, the value of this attribute equals the agnostic cost of the path from the advertising router to where the destination prefix attaches to the ISP network. For routes received from other ISPs, the value equals the sum of the agnostic cost advertised by the other ISP, after normalization (see below), and the agnostic cost of the internal path.

- Border routers of an ISP, i.e., routers that connect to other ISPs, keep track of the sum of incoming and outgoing agnostic costs for each neighboring ISP. This is implemented using two counters per neighbor that are updated each time a routing message that announces a new destination or withdraws an existing one is exchanged.

- Periodically, with a period of 30 seconds in my implementation, each border router shares its sums with the other border routers of the same ISP, which enables the computation of the normalization factors with each neighboring ISP. Border routers share a list of triplets, one for each neighboring ISP, containing the neighbor's AS number and the sums of incoming and outgoing costs. Information from all the routers is aggregated to compute the normalization factor, which is the ratio of the incoming to outgoing costs summed across all border routers.

  All ISPs with multiple border routers have a mechanism, such as a mesh of border routers (iBGP mesh [95]) or route reflection [15], for the border routers to be able to exchange information. I leverage this mechanism for sharing the sums of agnostic costs.

- Costs received from an ISP are normalized by multiplying them with the normalization factor for that ISP. In my implementation, this is done by the border router that received the route directly from the neighboring ISP; this router then propagates the normalized cost to the other border routers. An alternative is to have border routers normalize the cost independently, as all of them know the current normalization factor. While the former requires border routers to re-send routing messages when the normalization factor changes, it is more robust to transient inconsistencies in the normalization factor across the routers: because only one router is responsible for normalizing the costs of an incoming route, normalization factor inconsistencies do not lead to forwarding path inconsistencies.

- The path selection criteria for selecting the best route to a destination when multiple routes are present is slightly different from that of BGP. Table 4.1 compares the decision process of BGP and Wiser. The latter includes one additional step that selects routes based on the Wiser cost, which is the sum of the normalized received cost and the internal cost to the egress router. The decision process of Wiser, in a manner sim-

Table 4.1: Interdomain routing decision process with BGP and Wiser. Each step is a filter that selects a subset of the available routes, and successive steps are taken until only one route is left. The only difference between BGP and Wiser is that the latter uses a filter based on the Wiser cost of the received routes as the second step.

BGP	Wiser
1. Highest local preference	1. Highest local preference
2. Shortest AS-path length	**2. Lowest** Wiser **cost**
3. Lowest origin type	3. Shortest AS-path length
4. Lowest MED (with same next-hop AS)	4. $\cdots$
5. eBGP-learned routes over iBGP-learned	5. $\cdots$
6. Lowest IGP cost to egress router	
7. Lowest router ID of the BGP speaker	

ilar to that of BGP, attempts to strike a balance between internal cost and end-to-end path quality instead of exclusively focusing on internal cost. With BGP, ISPs prefer paths with shorter AS-path lengths over those with lower internal cost (though AS-path length can be a poor indicator of path quality in practice [118]). Similarly, with Wiser, ISPs prefer paths with lower Wiser cost over those with lower internal cost.

An essential requirement for a routing protocol is that the paths be free of loops. This property holds for Wiser. Routing path loops involving multiple ISPs will not form because, as is done in BGP, Wiser uses AS-path information to disallow such paths. Routing path loops internal to an ISP will not form if the ISP's internal routing produces loop-free routing because the Wiser cost of a route is simply the sum of its internal cost to the egress and the received cost which does not change within the ISP network.

- When the normalization factor for a neighboring ISP changes, the border routers connected to that ISP re-evaluate routes received from that ISP because routes that were previously not selected by the decision process may now be the best routes, or those that were previously selected may not be the best routes anymore. This re-evaluation is done in the background, while other tasks continue to be processed with a higher priority, because it can involve significant processing if many routes have been received from this neighbor. This re-evaluation is similar to what happens in BGP today when IGP costs change. However, while changes in IGP cost can lead to transient inconsistencies in forwarding paths computed by different routers [36], normalization factor changes do not lead to such inconsistencies, even if the re-evaluation proceeds at a different pace at border routers with different capabilities. This is because, as mentioned above, only one border router is responsible for normalizing the cost of an incoming route.

- Finally, to be able to verify a neighboring ISP's path selection behavior, border routers log information required to compute the incoming payment ratio. This information includes the amount of incoming traffic and the announced cost of each destination prefix. The amount of traffic received for each prefix is estimated using a sampling mechanism that is similar to Cisco NetFlow. Periodically, for instance, every five minutes, the estimates are logged to disk and reset. When an ISP wants to verify the behavior of its neighbor, it collects this information from all border routers and checks if the payment ratio, which is computed as the average payment divided by the average cost, is below the agreed upon threshold. Similar logging is implemented to compute outgoing payment ratios, which helps cross-checking if a neighboring ISP claims that this ISP has exceeded the payment ratio threshold.

Chapter 5

EVALUATION I: EFFICIENCY, OVERHEAD AND ROBUSTNESS TO CHEATING

I divide the evaluation of Wiser across two chapters. This chapter quantifies the efficiency, overhead and robustness to cheating of Wiser. In the next chapter, I explore the design space of routing protocols based on win-win coordination to understand, for instance, what aspects of the design of Wiser are essential to its efficiency.

I answer the following questions in the context of topologies, workloads, and ISPs' behaviors that I study.

1. How efficient is Wiser**?** I show that the efficiency of Wiser comes close to that of socially optimal routing when ISPs use comparable metrics, and of Pareto-optimal routing when ISPs use incomparable metrics. I also show that while the efficiency of *anarchy* is acceptable on average, it is poor in a small subset of cases. In today's Internet, fixing this poor tail requires manual intervention, leading to poor reliability and high operational cost. By contrast, the efficiency of Wiser is high even for the tail.

2. What is the overhead of running Wiser **compared to running BGP?** I quantify the overhead of Wiser across several dimensions of interest. I show that the implementation complexity and routing message processing requirements of Wiser are similar to BGP. For normal routing workloads, the computational overhead of Wiser is within 15 to 25% of BGP. Wiser has acceptable convergence time even in response to major routing changes.

3. How robust is Wiser **to cheating ISPs?** As argued in Section 4.5, while cheating is not expected to be commonplace, Wiser needs to be robust to the rare instances of cheating that may occur. The cheaters should not be able to gain much and the loss to honest ISPs should be small. I show that these properties hold for Wiser because of the cost normalization and payment ratio constraints on routing.

Below, I first provide an overview of the evaluation methodology and then present individual experimental results.

5.1 Experimental Methodology

The answers to the above questions depend on many aspects of ISP networks, some of which are hard to model. To focus on realistic rather than theoretical best- or worst-case bounds, I combine measured data with models based on known properties of the Internet. As measured input, I use a dataset of 65 measured ISP topologies and their interconnections [109]. These ISPs are diverse in terms of their sizes and geographical presence. A node in an ISP topology corresponds to a city where the ISP has a point-of-presence (PoP) and the links correspond to inter-city connections. The topologies are annotated with geographic coordinates of PoPs. The models that I use depend on the specifics of the experiment but a common one is approximating propagation delay of a link using the geographic distance between the two end-point cities. Prior work has shown this to be a good approximation [86].

For evaluation, I use a custom idealized simulator that does not model message passing, and two implementations of Wiser, one in a message-level simulator [113] and another in a router platform [129]. Because these three engines model different levels of detail, they have different scaling properties. The engine that I use for an experiment depends on the required level of detail and scale. The idealized simulator is the most scalable, and I use it for evaluating efficiency and robustness to cheating. I use the implementations for evaluating overhead.

The following subsections provide an overview of the experiments in this chapter.

5.1.1 Efficiency

The efficiency of global routing can be measured in several ways. I study two types of scenarios, where ISPs have the same optimization objectives and where they have diverse objectives. For the former, motivated by current problems with Internet routing [105, 62, 63], I explore two metrics of interest to ISPs and users. First, I consider in Section 5.2.1 the end-to-end length of Internet paths, which directly impacts not only application performance but also resource usage inside ISP networks, since longer paths consume more resources. This metric assumes that the network capacity is well-matched to the traffic it carries, and hence the ISPs are primarily interested in minimizing the lengths of paths inside their networks. Second, I consider in Section 5.2.2 the extent of bandwidth provisioning required inside ISP networks to minimize the possibility of overload when the traffic is no longer well-matched to the network capacity, for example, due to a failure. Finally, I consider in Section 5.3 the scenario of diverse ISP objectives.

I compare the efficiency of Wiser to that of *anarchy* and *optimal* routing. *Anarchy* refers to routing that is governed by current common practices. While deviations from these practices do exist, for instance, due to manual tweaking by operators, measurements show that current Internet routing is dominated by them [109, 124]. *Optimal* routing globally minimizes the metric of interest.

5.1.2 Overhead

I investigate in Section 5.4 the overhead of Wiser, relative to BGP, along several dimensions, including implementation complexity, convergence time, routing message processing and computation requirements. Towards this purpose, I implemented Wiser in SSFNet [113] and XORP [129]. SSFNet is a scalable, message-level simulator for network protocols and XORP is a flexible router platform. Both implementations follow the description in Section 4.6 except that the XORP implementation does not include virtual payment logging.

The need for two independent implementations is necessitated by the scope of the evaluation. Experimenting with large, realistic ISP topologies was not possible with XORP due to the lack of a suitable hardware infrastructure. For instance, emulating a 300-node topology requires 300 machines with controlled delay between each pair of machines that are connected in the topology. Since metrics such as routing message overhead are of interest only for such topologies, I cannot use XORP to study them. On the other hand, I cannot study the computational overhead using SSFNet, as detailed models of computation inside a router are not available in the simulator. Thus, I use both implementations to evaluate the entire range of metrics of interest.

5.1.3 Robustness to cheating

An ISP can have any number of motivations to cheat. While it is not possible to enumerate and study all of them, to understand Wiser's robustness to cheating, I consider in Section 5.5 two motivations that are perhaps most likely and have the potential to greatly impact honest ISPs. First, an ISP might want to reduce the average cost for the traffic it already carries. Second, if an ISP sells transit service and has a sufficiently well-provisioned network, it might want to attract more traffic to its network in order to gain more revenue.

5.2 Efficiency with Similar ISP Objectives

I start the empirical evaluation of Wiser by studying its efficiency when ISPs use similar optimization objectives. I consider two metrics of efficiency in this section. The first is based on the length of routing paths and the second is based on the amount of bandwidth provisioning required inside ISP networks.

5.2.1 Path length

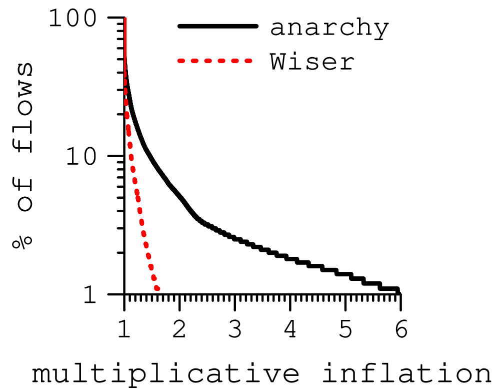
Paths in the Internet can be much longer than necessary [105, 109], which not only degrades application performance but also reduces network reliability because fixing overly long paths requires manual tweaking by operators. In this section, I evaluate how well Wiser optimizes such paths.

Methodology

I compute the lengths of routing paths produced by three different routing methods. The length of a path is the sum of the lengths of individual links, where I approximate link length using geographical distance. The routing methods that I study besides Wiser are *optimal* and *anarchy*. *Optimal* globally minimizes path length using information on the lengths of network links. It is impractical in the Internet context because it does not preserve ISP autonomy; I study it for comparative purposes. *Anarchy* mimics current interdomain routing using the common policies of shortest AS-path and early-exit. Thus, with *anarchy*, ISPs ignore the costs inside the neighboring ISPs' network. With Wiser, ISPs use internal distance as the basis for assigning costs to internal paths. This is a rough measure of the resources consumed inside the network, and minimizing it allows a smaller, thinner network to support a given amount of traffic. Following common commercial ISP policies, all three routing methods are constrained to paths that prefer customers, peers and providers, in that order, and do not provide transit to peers and providers [45]. To assign relationships to pairs of ISPs, I map ISPs to tiers [116], and I assume that ISPs at the same tier are peers and lower tier ISPs are providers of higher tier ISPs. In this experiment, I consider the entire internetwork of 65 ISPs. The traffic consists of a flow between each pair of PoPs.

Results

I find that the efficiency of Wiser comes close to that of *optimal* for this combination of topology, workload and ISP costs. The average path length with Wiser is only 4% higher than that with *optimal*. *Anarchy* does less well, with its average path length 13% higher than that of *optimal*. While this improvement might be important in some instances, it is



%	Anarchy	Wiser
50	1.0	1.0
10	1.4	1.1
5	2.0	1.2
1	5.9	1.5

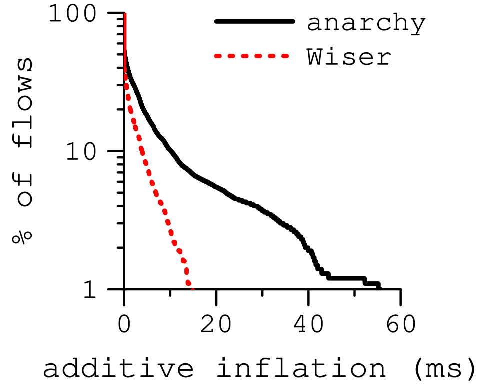
Figure 5.1: Multiplicative inflation in path length with *anarchy* and Wiser relative to *optimal*. The graph plots the CCDF of inflation on log scale. The table summarizes the graph.

probably not significant for most applications. This suggests that the common case path length in the Internet today is acceptable.

The key difference between *anarchy* and Wiser manifests itself in the distribution of path lengths for individual flows. Figure 5.1 shows the path length inflation with Wiser and *anarchy* compared to *optimal*. An inflation of two implies that the path length of the flow was doubled. The graph is the complimentary cumulative distribution function (CCDF), which means that the y-value represents the percentage of flows that were inflated by at least the corresponding x-value. The table summarizes the graph by listing a few points of interest.

The figure shows that while half of the paths are not inflated at all, some paths are highly inflated: 5% of the paths are inflated by at least a factor of two and 1% of them by nearly a factor of six. Applications using these paths will suffer very high latencies unless operators manually fix such paths.

Given that excessively poor performance in the tail increases the operational cost of today's Internet, the goal of ISP coordination is to optimize that tail. The figure shows that Wiser effectively meets this goal. While the top 1% of the flows were inflated by a factor of six with *anarchy*, they are inflated by only a factor of 1.5 with Wiser.



%	Anarchy (ms)	Wiser (ms)
50	0	0
10	9	4
5	22	7
1	55	14

Figure 5.2: Additive inflation in path length with *anarchy* and Wiser relative to *optimal*. The graphs plots the CCDF of inflation on a log scale. The table summarizes the graph.

The results above are obtained after adding 1 millisecond (ms) to all paths to ignore small improvements for very short paths and to simulate the distance of the end host to its nearest PoP. The results without adding 1 ms to paths are qualitatively similar but have a longer tail: the worst 1% of the paths are inflated by a factor of 7.2 with *anarchy* and 1.7 with Wiser.

Figure 5.2 shows the same data for another relevant measure of inflation. It plots additive inflation, or the additional length flows traverse with *anarchy* and Wiser, compared to *optimal*. Inferences similar to those above can also be made from this figure. It confirms that high inflation is not limited to short paths, for instance a 1 ms path becoming 2 ms, but leads to longer absolute path lengths as well.

These results have two important implications. First, Wiser can achieve efficient routing in the Internet, marginalizing the need for manual intervention. Second, since the bilateral barter of Wiser comes so close to *optimal*, (potentially more complicated) approaches based on multilateral barter or global currency are unwarranted for the topology and workload that I consider.

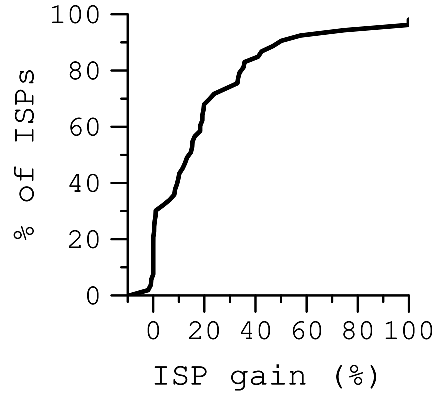


Figure 5.3: The CDF of gain for individual ISPs with Wiser, measured as the reduction in average distance relative to *anarchy*.

Win-win routing I now show that improvement in end-to-end path length with Wiser, compared to *anarchy*, does not come at the cost of individual ISPs suffering for the global good. This win-win property incents ISPs to adopt the protocol. I measure the gain of an ISP as the average reduction in distance, relative to *anarchy*, that a packet travels inside the ISP's network with Wiser. This measure of gain is consistent with the internal optimization metric used by the ISPs in this experiment, but it does not account for the end-to-end performance improvement experienced by the ISP's customers or the savings in operational cost. As such, it is a lower bound on the gain an ISP will experience with Wiser.

Figure 5.3 plots the CDF of gain for individual ISPs with Wiser. It shows that almost no ISP loses, rather than some ISPs gaining and others losing. Even when an ISP's internal cost stays the same, it has an incentive to adopt Wiser if that leads to better performance for its customers.

Unlike Wiser, *optimal* routing does not account for the gain or loss experienced by individual ISPs, and can thus result in individual ISPs losing for the greater good. While this difference is not evident in the overall Internet topology, because the net gain is high, it manifests itself in smaller topologies. To demonstrate this point, I consider topologies

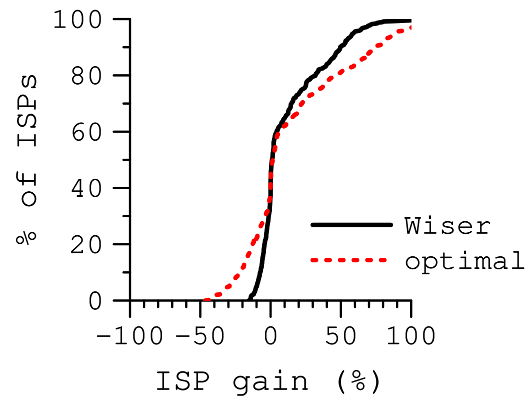


Figure 5.4: The CDF of gain for ISPs in individual bilateral relationships with Wiser and *optimal*. The gain is measured as the average reduction in distance relative to *anarchy*. There are two points in each CDF for each pair of ISPs.

that consist of pairs of ISPs with two or more interconnections so that there are multiple interdomain routing choices. Figure 5.4 plots the gain for individual ISPs within ISP pairs. There are two points for each pair of ISPs. It shows that individual ISPs can sometimes suffer significant losses with *optimal*. With Wiser, losses are smaller and are experienced only by a small fraction of ISPs. On closer inspection, I find that these ISPs correspond to cases where there is not much benefit from coordination. Such ISPs can revert to *anarchy* without significantly impacting efficiency. But I argue that even these ISPs have an incentive to adopt Wiser because the losses are small and adopting Wiser will automatically improve any egregiously bad paths.

5.2.2 Bandwidth provisioning

I now consider another ISP objective for the scenario where ISPs have similar objectives. The objective is reducing bandwidth provisioning. To operate a congestion-free network in the face of dynamic conditions, ISPs can either choose a high provisioning level, precluding congestion for most load variations, or choose a combination of a lower provisioning level with dynamic congestion alleviation. The former approach is more common today because BGP does not enable ISPs to stably react to load variations. Wiser enables ISPs to lower the provisioning level by dynamically reacting to congestion. I quantify this benefit in this section.

Methodology

Network provisioning is harder to evaluate than path length because it is affected by more factors, such as link capacities and workloads, and information on these factors is not available. To obtain results that are representative of the Internet, I approximate these factors using known properties of Internet routing.

First, to approximate the amount of traffic between a source and destination PoP, I use a gravity model [77, 132]. This model states that the amount of traffic between a pair of PoPs is proportional to the product of the "weight" of the PoPs. As the weight of a PoP, I use the population of its city, estimated as the number of people in a 50×50 square mile grid centered on the geographical coordinates of the city. I use population density data from CIESIN [28] for this purpose. This workload model leads to a skewed traffic matrix with larger cities consuming more bandwidth, both hallmarks of real Internet traffic [66, 17].

Second, to model link capacities, I assume that capacity is proportional to the stable load on the link, i.e., in steady-state a well-designed network tends to be roughly matched to its traffic so that links that carry more traffic tend to be of higher capacity [132]. The traffic matrix combined with the routing within an ISP enables the computation of the load on each link. To preclude results being dominated by links that carry little traffic, I assume that the base capacity of all links that carry less than the median load for an ISP is the median itself; these would then be the underutilized links in the topology.

Third, motivated by congestion that occurs when interconnections between large ISPs have problems today [62, 63], I simulate dynamic conditions by modeling the failure of interconnections between tier-1 ISPs. Because these are infrequent but major events in the

Internet, an ISP that is equipped to deal with them should be able to handle most other events.

Finally, I measure efficiency using the overprovisioning level. The overprovisioning level for a link is the additional capacity, compared to its stable load, required to support all simulated failures. For instance, if the maximum load on a link across all failures is twice that of the stable load, the overprovisioning level is 100%. The overprovisioning level for an ISP is the weighted average of the overprovisioning levels of its links. The stable load on the link is used as its weight, which reflects the economic reality that doubling capacity is costlier for links with higher capacity.

I compare the overprovisioning level for three different routing behaviors.

1. *Optimal* assumes that the internetwork is one ISP and minimizes its overprovisioning level. It is computed by solving a linear programming problem [81] using *lp_solve* [16]. Because of computational limits, I compute this only over subsets of the overall topology (see below). I allow fractional routing, which means that traffic between two nodes in the topology can use multiple paths, with an arbitrary fraction of the traffic traversing each path. Fractional routing provides a lower bound on the overprovisioning as routing in the Internet is usually non-fractional.

2. With Wiser, ISPs will choose to assign load-dependent costs to their links in a manner that depends on their network. I experiment with a simple method, shown in Figure 5.5, in which the cost of a link increases roughly linearly with load. The magnitude of cost increment is controlled by the quantization threshold, t, which is a measure of how closely cost and link utilization are coupled. Higher values imply that the link can tolerate greater deviations in load. I experiment with different values of t. When the load is below the "stable load" (reflective of capacity), the cost of a link is the same as that used for the path length experiments in the previous section. Beyond that the cost changes when the current load and cost factors, relative to their stable values, are offset by more than a factor of t. It increases in steps of t and de-

```
costFactor[i] = linkCost[i] / stableLinkCost
loadFactor[i] = linkLoad[i] / stableLinkLoad
if loadFactor[i] > costFactor[i] + t then
   costFactor[i+1] = costFactor[i] + t
else if loadFactor[i] < costFactor[i] - t AND costFactor[i] ≥ 1 + t then
   costFactor[i+1] = costFactor[i] - t/10
else
   costFactor[i+1] = costFactor[i]
end if
linkCost[i+1] = costFactor[i+1] × stableLinkCost
```

Figure 5.5: The algorithm for changing link costs based on load. It shows how the link cost for the next iteration $(i + 1)$ is derived from the load and link cost in the current iteration (i). The quantization threshold, t, controls how closely link cost is coupled to link load.

creases in steps of $t/10$. A bigger cost increment enables overloaded links to quickly disperse some of their traffic. The experiment proceeds in iterations. In an iteration, the cost of each link is determined using the method shown in the figure, and the new routing pattern is computed based on those costs. The experiment terminates when none of the link costs change.

Admittedly, the method above for varying cost with load is crude; it is only intended to illustrate the benefit of Wiser with dynamic cost assignment. Several other methods are possible. For instance, to hide internal changes when a link becomes overloaded, an ISP can first try to reroute traffic within its own network to handle the overload; if that proves insufficient, it can then modify the costs it announces to other ISPs. As another possibility, instead of changing the costs for all traffic traversing a link, ISPs can change the costs for only popular destinations.

3. *Anarchy* simulates currently common interdomain routing practices. It is load-insensitive and computed as in the previous section, after removing any failed links from the topology. While it is possible for ISPs today to respond to overload, it is not common because the response can lead to instability, as illustrated by the example in Section 2.2. For simplicity, I do not model ISPs responding in a way that does not impact other ISPs; while such techniques exist, their efficacy in response to major changes is limited in practice [37]. Thus, my model of anarchic routing measures the overprovisioning level required to stably withstand failures without any coordination.

Results

I divide the results into two parts. I first use subsets of the overall topology and compare all three routing methods. This yields a measure of how close to *optimal* Wiser can get. I then use the entire internetwork topology, for which *optimal* could not be directly computed because of computational limits, and compare only Wiser and *anarchy*.

I first consider topologies for which the *optimal* could be computed. These consist of pairs of neighboring ISPs in the dataset, with traffic flowing between all pairs of PoPs in the two ISPs. I restrict this experiment to ISP pairs that interconnect in three or more places before the failure so that there are at least two routing choices after the failure. With Wiser, the value of the quantization threshold, t, is 10%.

Figure 5.6 shows the overprovisioning level required to deal with the failures that I simulate. The left graph shows the overprovisioning for individual links. A point in this graph corresponds to a link in a two-ISP topology. The overprovisioning level is zero for more than half of the links because most links are on the edges and do not carry any additional traffic as a result of failures. The right graph shows the overprovisioning for ISP networks. There are two points in this graph for each ISP pair.

The graphs show that Wiser closely approximates the *optimal* and that the overprovisioning level with *anarchy* is higher. Relative to the *optimal*, the median ISP-level overprovisioning is 0% with Wiser and 7% with *anarchy*. Even though ISPs overprovision

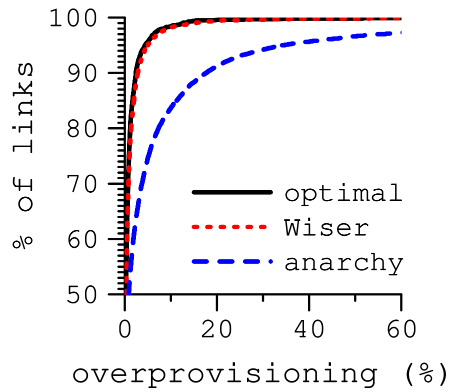


Figure 5.6: Overprovisioning required with different routing methods to deal with interconnection failures between pairs of ISPs. *Left:* The CDF of link-level overprovisioning. *Right:* The CDF of ISP-level overprovisioning; there are two points for each ISP pair. Note that the x- and y-axis ranges are different in the two graphs.

whenever they upgrade their networks, this difference in bandwidth provisioning would translate into significant monetary savings for ISPs if they need to upgrade less often.

I now consider the complete internetwork topology in the dataset and show results for Wiser and *anarchy* (since computing *optimal* was computationally intractable). Traffic in this experiment consists of flows between a randomly selected 10% of all possible PoP pairs. The flow sizes are computed using the gravity model, as explained above. I simulate the failure of each interconnection between tier-1 ISPs. There are over 400 such interconnections in the dataset.

Figure 5.7 shows the results, with the link-level view on the left and the ISP-level view on the right. The value of the quantization threshold, t, is 10%; I investigate other values below. The x-axis range for ISP-level overprovisioning is smaller in this figure than that in Figure 5.6 because the failure of an interconnection in the entire topology is a smaller relative change.

As is the case for the smaller topologies, the overprovisioning with Wiser is much less

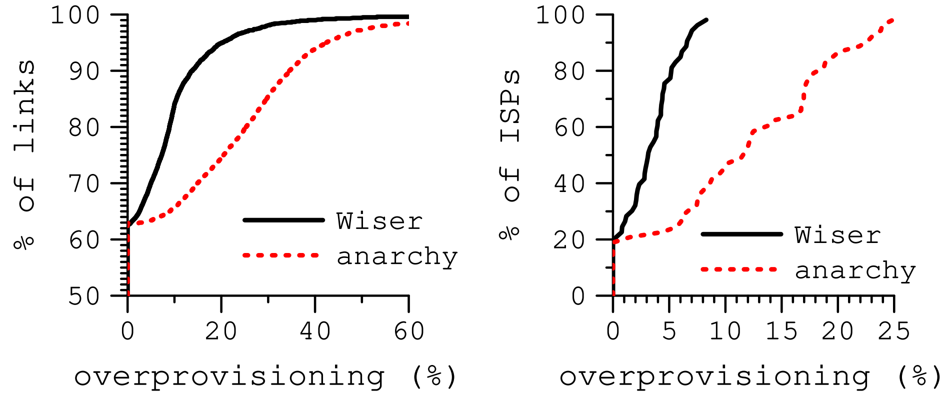


Figure 5.7: Overprovisioning required with Wiser and *anarchy* to deal with tier-1 interconnection failures. *Left:* The CDF of link-level overprovisioning. *Right:* The CDF of ISP-level overprovisioning. Note that the x- and y-axis ranges are different for the two graphs.

than that with *anarchy*. For individual links, the difference in the 90th percentile overprovisioning level between *anarchy* and Wiser is 20%. For ISPs, the difference at the 90th percentile is 16%, and the average difference is 8%. This difference can lead to significant monetary savings if Wiser requires ISPs to upgrade their network less often.

Impact of quantization threshold I now study the impact of quantization threshold, t. Figure 5.8 plots the overprovisioning for different values of the threshold. The curve for *anarchy* is reproduced for comparison. Because the simulations are often not stable with t=1%, the results shown here are obtained by terminating them after twenty cost changes for any link. The instability underscores that managing overload is easier when a network is overprovisioned to some extent such that the links can tolerate small increases in load. This is because overprovisioning makes it easy to disperse load on a congested link without overloading other links.

The graph shows that a threshold of 10% does almost as well as a threshold of 1%. The overprovisioning for a threshold of 20% is higher but still better than *anarchy*. These results

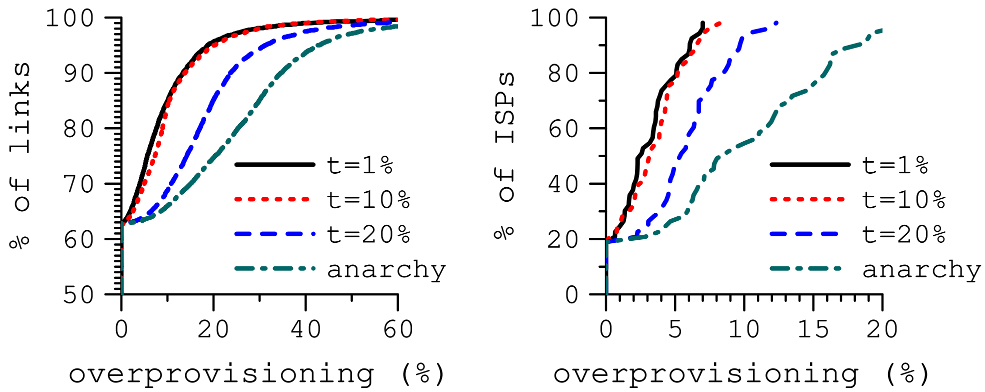


Figure 5.8: The impact of quantization threshold on overprovisioning. The curves for t=10% and *anarchy* are same as those in Figure 5.7. *Left:* The CDF of link-level overprovisioning; note that the y-axis starts at 50%. *Right:* The CDF of ISP-level overprovisioning. Note that the x- and y-axis ranges are different for the two graphs.

suggest that, as argued in Section 4.4, a coarse dependence of link costs on utilization is sufficient for networks that are well-engineered to deal with a broad class of load variations. Overprovisioning required to operate a congestion-free network is almost the same as the case where the costs are tightly coupled with utilization.

Win-win routing I conclude this section by evaluating whether Wiser is win-win from a bandwidth provisioning perspective as well. Figure 5.9 plots the gain for individual ISPs, measured as the overprovisioning required with *anarchy* minus that with Wiser ($t = 10\%$). None of the ISPs lose with Wiser and a third of them gain more than 10%.

5.3 Efficiency with Heterogeneous ISP Objectives

So far I have considered scenarios where ISPs have comparable objectives, but a more realistic case is when the ISPs have diverse objectives. I now investigate the efficiency of Wiser in such scenarios. I first consider in Section 5.3.1 a particular model of diverse ISP objectives, which uses link weights that are consistent with routing observed inside ISPs.

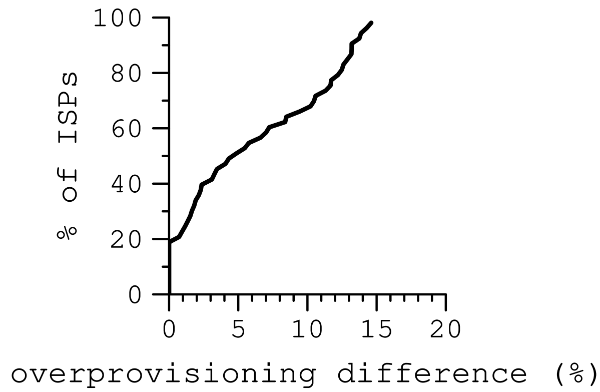


Figure 5.9: The CDF of the difference in overprovisioning between *anarchy* and Wiser for individual ISPs.

Then, I use constructive models in Section 5.3.2 to gain further insight into the efficiency of Wiser with diverse ISP objectives.

5.3.1 Inferred link weights

Mahajan *et al.* propose a method to infer the link weights of an ISP topology by observing the routing paths used inside it [72]. Shortest path routing produced by these weights is consistent with the observed routing, though these weights are not necessarily what the ISP itself uses. I assume that these weights model the ISPs' objectives and study the efficiency of Wiser in this scenario.

I infer weights using the same routing path data as that used by Spring *et al.* [109] to measure the ISP topologies in my dataset. Routing with Wiser and *anarchy* is computed in a manner similar to path length, except that ISPs optimize for their inferred link weights rather than distance.

Figure 5.10 shows the efficiency of Wiser and *anarchy* with inferred link weights. It plots the multiplicative inflation in path length relative to *optimal* that was computed in Section 5.2.1. The graph shows that Wiser comes close to *optimal* for inferred link weights.

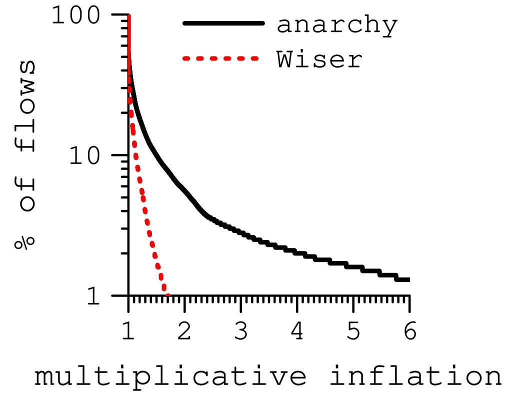


Figure 5.10: Efficiency of Wiser and *anarchy* with inferred link weights. The graph plots the CCDF of multiplicative inflation in path length relative to path length with *optimal*.

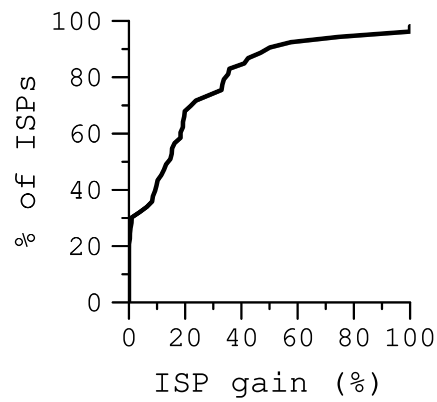


Figure 5.11: The CDF of gain for individual ISPs with inferred link weights. The gain for an ISP is measured as the reduction in average weight relative to *anarchy*.

Figure 5.11 shows that Wiser is win-win when ISPs use inferred link weights. It plots the CDF of gain for individual ISPs. The gain for an ISP is measured as the reduction in average weight of carrying traffic inside it with Wiser, relative to *anarchy*. No ISP loses with Wiser.

5.3.2 Other objectives

The last section studied the efficiency of Wiser for a particular model of ISP objectives; in this section, I use a constructive model to further understand the efficiency of Wiser when ISPs use different objectives.

ISPs in the Internet use a variety of optimization metrics, and even if a complete list of possible ISP objectives were available, simulating all possible combinations is not tractable. I simulate a simpler alternative in which I assume that each ISP has an unknown (to me) objective, and randomly assign a cost to each link in the range [0..1]. In reality, different ISPs will have different ranges but the cost normalization in Wiser implies that the absolute range used in the experiment is not important to the resulting routing. Routing with Wiser and *anarchy* is computed in a manner similar to path length, except that ISPs optimize for these costs rather than distance.

When ISPs have incomparable objectives, Wiser aims to approximate Pareto-optimality. There are multiple Pareto-optimal solutions in the system, and verifying whether a given routing pattern is Pareto-optimal is computationally hard. Instead, I compare the efficiency of Wiser to a Pareto-optimal routing pattern that minimizes the sum of total costs that each ISP incurs for carrying traffic. Even though the costs of ISPs are not directly comparable, minimizing the sum of ISPs' costs leads to a Pareto-optimal routing pattern. If Wiser is close to this pattern, its efficiency can be considered to be close to Pareto-optimal.

Figure 5.12 shows the efficiency of Wiser and *anarchy* compared to this routing pattern by plotting the CCDF of multiplicative inflation in flow costs. The cost of a path is the sum of the costs of the constituent links. The inflation is measured relative to the cost of the flow in the chosen Pareto-optimal routing pattern. Unlike *anarchy*, Wiser is close to the chosen Pareto-optimal routing pattern: 60% of the flows are not inflated at all, 5% of the flows are inflated by less than a factor of 1.5, and 1% of the flows are inflated by less than a factor of 2.5. In contrast, with *anarchy*, 5% of the flows are inflated by a factor of 3, and 1% of them are inflated by a factor of 4.9.

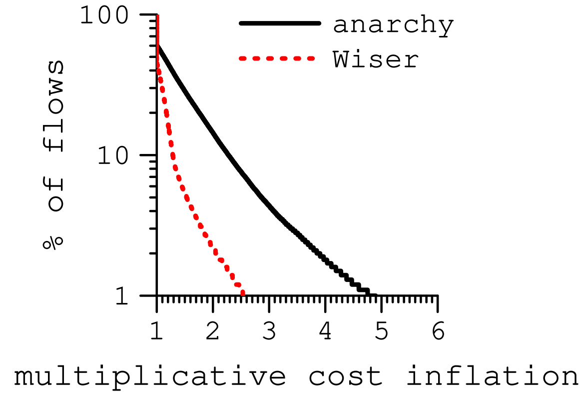


Figure 5.12: Efficiency of Wiser and *anarchy* with heterogeneous ISP objectives. The graph plots the CCDF of multiplicative inflation in cost relative to the chosen Pareto-optimal routing pattern.

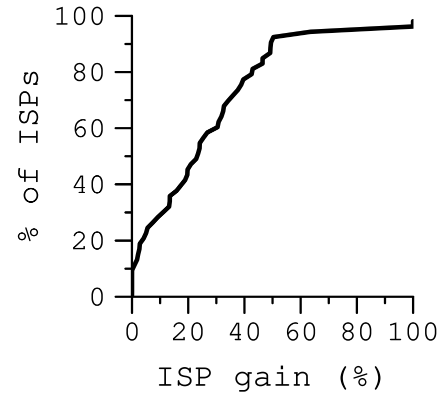


Figure 5.13: The CDF of gain for individual ISPs with heterogeneous ISP objectives. The gain for an ISP is measured as the average reduction in cost relative to *anarchy*.

To evaluate whether Wiser is win-win for this scenario of diverse ISP objectives as well, I measure the gain for individual ISPs as the reduction in average cost of carrying traffic with Wiser, relative to *anarchy*. Figure 5.13 shows that no ISP loses with Wiser. Thus, Wiser enables ISPs with diverse objectives to cooperate in a way that produces more efficient routing while ensuring that each ISP gains in terms of its own objective.

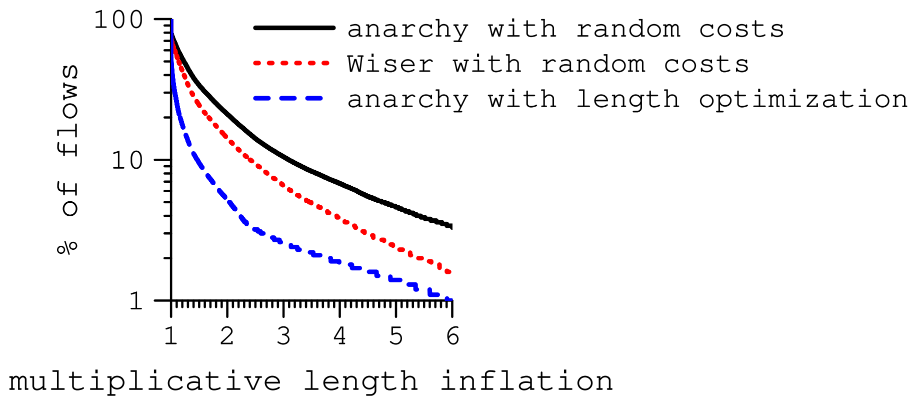


Figure 5.14: The CCDF of multiplicative inflation in path length with different routing methods. The inflation is measured relative to *optimal* that minimizes path length.

End-to-end path quality Enabling ISPs to optimize for their own objectives raises concerns regarding the quality of end-to-end routing paths, something of utmost concern to users. If ISPs' objectives do not compose well, end-to-end paths may be poor. Figure 5.14 shows that this is true for randomly assigned ISP costs. It plots the multiplicative inflation in path length, relative to *optimal* routing that minimizes path length (as in Section 5.2.1). For comparison, I also reproduce the curve for *anarchy* from Figure 5.1 in which each ISP routes flows based on internal length, not randomly assigned cost. While Wiser with random agnostic costs does slightly better than *anarchy* with random ISP costs, it is worse than *anarchy* in which each ISP optimizes for length.

However, ISPs' costs are not truly random but rooted in meaningful traffic and network optimization goals. (I studied random costs above to show that even when ISPs' costs have nothing in common, Wiser enables ISPs to approximate Pareto-optimal routing such that each ISP gains according to its own objective.) For instance, all reasonable ISP metrics are likely to reflect path length, a key measure of interest to users, though its contribution toward the cost metric might vary across ISPs. Thus, the resulting end-to-end routing paths can still be high quality because the ISPs' metrics will roughly compose even though

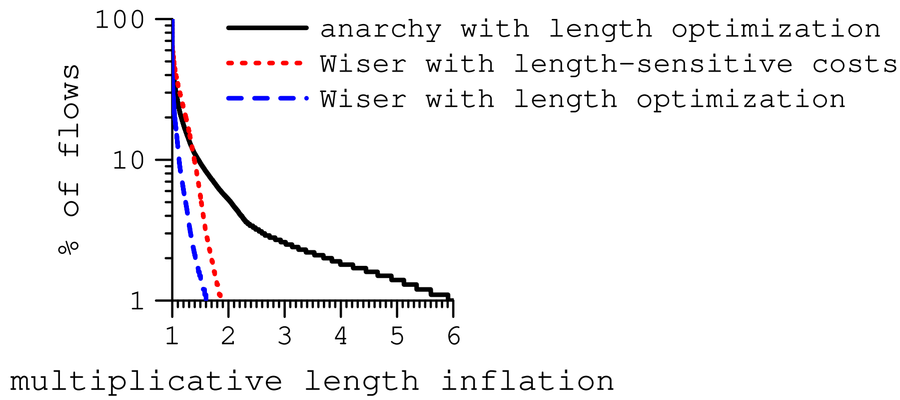


Figure 5.15: The CCDF of multiplicative inflation in path length with different routing methods. The inflation is measured relative to *optimal* that minimizes path length.

they are not identical. This is probably why the path lengths with inferred link weights in Section 5.3.1 are close to *optimal*.

To empirically evaluate the hypothesis above, I conduct an experiment in which ISPs' costs are not randomly assigned but are length-sensitive. The cost of a link is $l \times 0.5 + l \times r$, where l is the length of the link and r is a random number in the range [0..1]. $l \times r$ captures the unknown components in each ISP's objectives, scaled to match the contribution from the length component. Figure 5.15 shows the resulting end-to-end path length distribution. For comparison, I reproduce the two curves of Figure 5.1 in which ISPs' costs are completely determined by link length. Wiser with length-sensitive costs is significantly better than *anarchy* and only slightly worse than Wiser when ISPs optimize for distance.

5.4 Overhead

An important requirement for a practical routing protocol is low overhead. I next study the overhead of Wiser in terms of its implementation complexity, convergence time, routing message processing and computation requirements. I use implementations of Wiser in SSFNet and XORP and compare the overhead to that of BGP, the current interdomain routing protocol in the Internet. It is desirable that the overhead of Wiser be comparable to that of BGP.

5.4.1 Implementation complexity

As a rough measure of the implementation complexity of Wiser, I count the number of additional lines of code required to implement it, starting from an existing BGP implementation. The SSFNet BGP daemon consists of 26,000 lines of Java, and the Wiser implementation adds 1500, or roughly 6%, lines of additional code. The XORP BGP daemon consist of 32,000 lines of C++, and the Wiser implementation adds 1000, or roughly 3%, lines of additional code. The number of lines includes comments in each case, and the Wiser implementations have not been optimized to reduce the line count.

Perhaps because of XORP's focus on flexibility [50], implementing Wiser in XORP took 20% less time: 8 days, compared to the 10 days required for the SSFNet implementation. The number of days includes the time spent understanding the existing code base. The low implementation effort of Wiser for both implementations is a testament to its simplicity as a protocol.

5.4.2 Convergence time and routing message overhead

Next, I study the convergence time and routing message overhead of Wiser. These measures need to be studied in response to changes that perturb routing because both Wiser and BGP generate routing messages only when routing path changes are required. Convergence time indicates the time that it takes for the routing to fully react to the change; the routing message overhead indicates the load on the routers in terms of the number of routing messages that they need to process in the meantime.

I use SSFNet for these experiments. Due to memory limitations, I conduct experiments only over the "core" of the internetwork topology. This core includes all tier-1 nodes that have more than one neighbor in the dataset. There are roughly 300 such nodes. Because

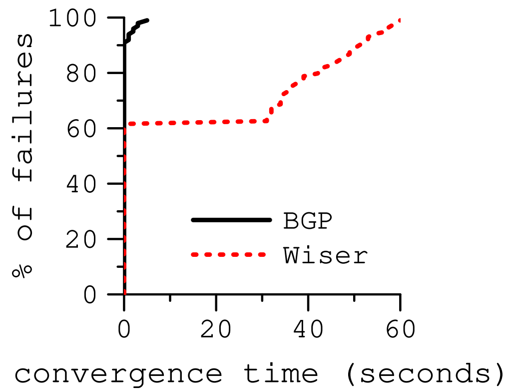


Figure 5.16: The CDF of convergence time of BGP and Wiser with load-insensitive agnostic costs.

the metrics that I study depend largely on the nature of this highly interconnected core [64], the results obtained over this subset should be reflective of the overall topology. The border routers within each ISP are organized in a fully-connected (iBGP) mesh [95].

The routing perturbations that I consider are the failures of the interconnections between tier-1 ISPs (as in Section 5.2.2). Each failure is a significant perturbation; most changes will have a smaller impact on routing. I study two scenarios: where ISPs use load-insensitive agnostic costs and where they use load-sensitive agnostic costs.

Load-insensitive agnostic costs

I first study the scenario where agnostic costs are independent of load. Because ISP networks are highly overprovisioned, I expect this to be the common case in the Internet. Costs are determined using geographic distances, as in Section 5.2.1.

Figure 5.16 shows the results for convergence time. It plots the CDF of the time it takes for the routing to converge after a link failure, where convergence is defined as the point at which no more routing changes happen. For a majority of the simulated failures, the convergence time of Wiser and BGP is similar. But for 40% of the cases, convergence

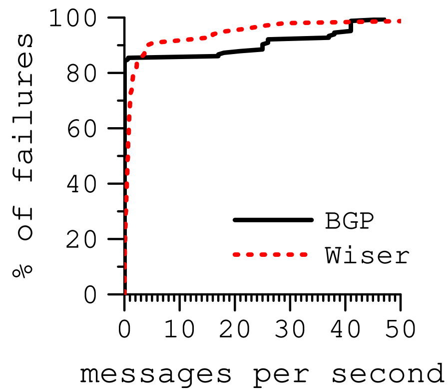


Figure 5.17: The CDF of maximum rate of routing messages for BGP and Wiser with load-insensitive agnostic costs.

time is higher for Wiser than for BGP. In BGP, routers advertise only reachability, and the routing converges as soon as the border routers attached to the failed link withdraw routes that use that link and announce new routes to their neighbors. But with Wiser, there can be a second round of route announcements if the normalization factor changes due to the link failure. However, the delay in the second round of advertisements is dominated by the MRAI (minimum route advertisement interval) timer of BGP, with a default value of 30 seconds. This timer determines the minimum gap between two routing updates sent to a neighboring router. Experiments confirm that lower values of this timer, as advocated by some researchers [48, 88], lead to faster convergence in Wiser. A possible way to decrease the convergence time of Wiser without altering the value of the MRAI timer is to selectively ignore MRAI when announcements are sent due to changes in the normalization factor; I have not experimented with such a mechanism yet.

Even the existing convergence time of Wiser should be acceptable in the face of major changes that I simulate, especially because the higher convergence time does not imply a higher path unavailability time. Connectivity in Wiser is restored as quickly as in BGP.

Figure 5.17 shows the results for routing message overhead. It plots the CDF of the

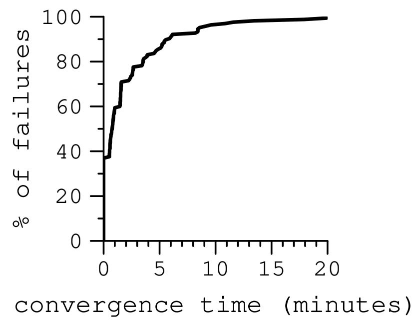


Figure 5.18: The CDF of convergence time of Wiser with load-sensitive agnostic costs.

maximum message rate experienced by any router in the topology. Message rate at a router is the number of messages received after the link failure divided by the time it takes for the routing to converge. The graph shows that Wiser and BGP have similar routing message overhead. The slightly lower message rate of Wiser is probably due to its higher convergence time for a subset of the cases.

Load-sensitive agnostic costs

I now consider the scenario where the agnostic costs are load-sensitive and study the convergence time and routing message overhead for Wiser alone, as there is no systematic way to achieve load-sensitive routing in BGP. The agnostic costs of ISPs are determined using the mechanism described in Section 5.2.2 except that there is a minimum time of two minutes between consecutive cost changes for a link to allow time for routing to fully react to the previous change.

Figure 5.18 plots the CDF of convergence time with Wiser. It shows that in 90% of the cases, routing converges in less than five minutes. But it can take up to 20 minutes in the extreme, which stems from my conservative approach to changing costs in the interest of stability. Stable and quickly-converging load-sensitive routing for large-scale interdomain networks is an open research question [125, 14, 106, 56], and future research will hopefully reduce this time. However, even this convergence time should be acceptable for significant

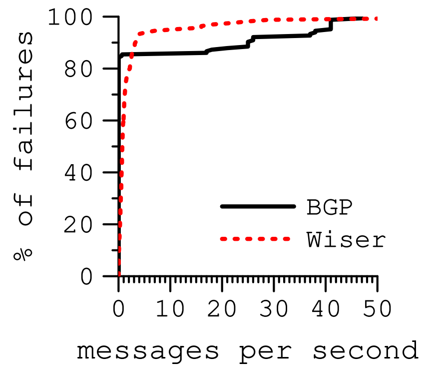


Figure 5.19: The CDF of maximum rate of routing messages for Wiser with load-sensitive agnostic costs. The curve for BGP is reproduced from Figure 5.17.

failures and is arguably much better than what human operators can achieve. As for the case of load-sensitive costs, connectivity is restored fairly quickly.

Figure 5.19 plots the CDF of maximum message rate at any router in the topology. For comparison, I reproduce the curve for BGP from Figure 5.17. The graph shows that even when load-sensitive costs are used, the routing message rate of Wiser is similar to that of BGP.

5.4.3 Computational overhead

As a final measure of the overhead, I study the computational load that Wiser incurs at the router. In addition to the load imposed by BGP, routers that run Wiser compute the sums of agnostic costs of incoming and outgoing routes, and also compute the normalization factor for each neighboring ISP. On the other hand, these routers will have a shorter decision process when they can select the best route in fewer steps using Wiser costs (Table 4.1), which reduces computational load. In this section, I study the combined effect of these activities on the computational requirements of Wiser.

I use XORP to quantify the computational overhead of Wiser. While computational load on a router is most interesting when the router is part of a large realistic topology, I cannot directly use XORP to emulate such topologies. To get around this issue, I collect logs of routing messages received by a router that is part of such topologies and feed this workload to a machine running XORP. The mechanism for collecting message logs is specified below. I conduct these experiments on a Linux 2.4.26 machine with a 2.2 GHz Intel Xeon processor and 3.8 GB of memory. The computational requirement of interdomain routing is measured using the Unix *time* command. I measure it in isolation of other protocols; the overall relative overhead on a router that switches from BGP to Wiser will be lower than that indicated by my results as routers usually run other protocols, such an intradomain routing protocol.

I measure the processing overhead of Wiser in two scenarios. The first corresponds to normal operating conditions for a router involved in interdomain routing in the Internet today. The second corresponds to highly dynamic conditions.

Normal operating conditions

To understand the processing overhead of Wiser under normal conditions, I feed in the log of routing messages at a RouteViews route server [103] that interconnects with forty-one diverse routers in the Internet. I use logs from two days, the 1st and 2nd of September, 2005. Because these messages do not contain Wiser costs, I attach a randomly generated cost in the integral range [1..10] to each message. To ensure that the routing tables fit in memory, I randomly select ten out of the forty-one message sources in an experiment. Different experiments have different sets of randomly selected sources. In all, I conduct five experiments for logs from each day.

I find that the computational load of Wiser is only 15 to 25% higher than that of BGP for the workload I study.

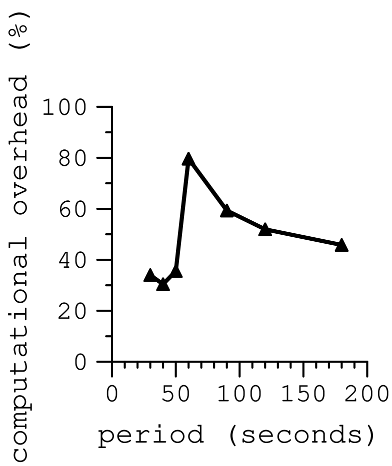


Figure 5.20: Computational overhead of Wiser relative to BGP, measured as the percentage increase in processor load. The x-axis represents the period with which links are failed in the topology.

Highly dynamic conditions

To understand the processing overhead of Wiser under highly dynamic conditions, I consider a scenario in which links inside tier-1 ISPs fail and recover continuously. Such extreme conditions are highly unlikely to arise in practice; this experiment is designed to stress-test Wiser.

I use SSFNet to generate a log of routing messages. The simulation topology consists of two tier-1 ISPs, AT&T and Sprint. Each node exports 300 prefixes to the rest of the internetwork. With this, the total number of prefixes crossing the AT&T-Sprint boundary is similar to the roughly 10K prefixes that Sprint announces to AT&T, as seen at a publicly accessible AT&T route server [102]. Wiser costs are determined by distance. Periodically, with different periods in different experiments, I fail a randomly selected link in the topology. The failed link recovers after one period, at which point a new link is failed. I log routing messages seen at ten randomly selected routers and feed them to the XORP test router.

Figure 5.20 shows the computational overhead of Wiser relative to BGP. Observing the graph from right to left, as the period is reduced, the overhead of Wiser increases as routers do more work to keep the normalization factor up-to-date and re-evaluate routing tables when the rate changes. Decreasing the period further reduces the overhead because the update frequency of the normalization factor is limited by the frequency with which border routers share their sums of incoming and outgoing agnostic costs. The graph shows that even at the peak of these highly dynamic conditions, the computational load of Wiser is less than twice that of BGP.

5.5 Robustness to Cheating

The experiments above show that Wiser enables efficient routing between cooperating ISPs and with low overhead, but to avoid potential abuse it must also be robust to cheating ISPs that deviate from the protocol specification. I study two likely motivations for cheating – reducing the average cost of the traffic that an ISP already carries and attracting more traffic to its network to get more revenue.

5.5.1 Reducing average cost

An ISP can try to reduce the average cost of the traffic that it carries using two methods, depending on whether the traffic is incoming or outgoing. For incoming traffic, it can be dishonest about the agnostic costs that it advertises to its neighbors such that the traffic enters its network at more favorable interconnections. For outgoing traffic, it can dishonestly select interconnections to neighboring ISPs. Below, I study the impact of each of these behaviors, first in isolation and then in combination.

I consider pairs of ISPs that interconnect in multiple places. This allows me to study in isolation the average cost of carrying traffic because the overall traffic entering and leaving the ISPs' networks stays constant. Traffic is composed of a unit flow between each PoP in one ISP to each PoP in the second.

I use distance as the ISP cost metric (as in Section 5.2.1). Another potential objective is one based on reducing traffic on overloaded links. But that is easily accomplished by (honestly) increasing the cost of the overloaded links to drive traffic away from them.

Incoming traffic

To investigate the robustness of Wiser to a cheating ISP that tries to unfairly reduce the cost of its incoming traffic, I simulate a situation in which the downstream ISP is dishonest about the costs it discloses to the upstream ISP. I assume that the dishonest downstream ISP has complete information about the traffic patterns and the other ISP's costs for sending traffic. This is an overestimation of the cheater's abilities because in practice there will be uncertainty in this information. The uncertainty can be increased, for instance, if the honest ISP randomizes its path selection among nearly equal cost paths. Using this information, the cheating ISP can compute the set of costs that will influence path selection in its favor. Cost normalization imposes a constraint on the set of dishonest costs, as they cannot be completely arbitrary.

Computing the dishonest costs to bring maximal gain to the cheater within the normalization constraint is NP-hard. Instead, I use a simple hill climbing algorithm to approximate such costs. The algorithm works in iterations. Each iteration starts with a random ordering of destination and interconnection point pairs. Considering each pair in order, the downstream ISP computes the cost (of that destination through that interconnection) that would bring maximal gain, using precise information about the traffic and the behavior of the upstream ISP. The algorithm ends when no cost changes occur in an iteration. I experiment with different seeds for the random number generator to ensure that the results are robust. I also obtain qualitatively similar results with an algorithm based on simulated annealing [60].

Figure 5.21 shows the results of this experiment. The top and middle graphs plot the CDF of the gain for the honest and the dishonest ISPs, where ISP gain is measured as the reduction in average distance relative to *anarchy*. The "no constraint" curve corresponds to a situation where the normalization constraint on downstream ISP's costs does not exist, giving it the ability to advertise arbitrary costs. The graphs show that, for the strategy I consider, both the loss for the honest ISPs and the gain for the dishonest ISPs are small with Wiser. Both the loss and gain is significant in the absence of the normalization constraint, which underscores the value of normalization in making Wiser robust to cheating.

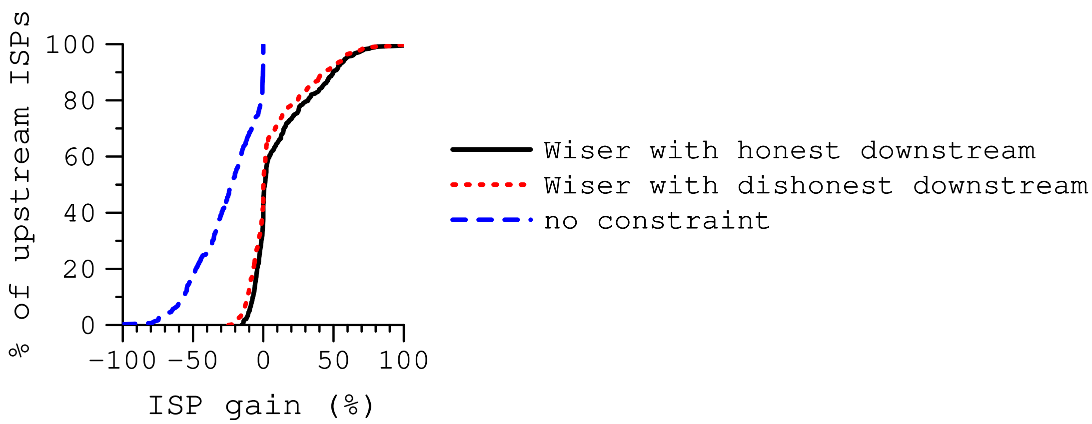
The bottom graph in Figure 5.21 shows the impact of cheating on traffic performance. It plots the multiplicative inflation, relative to *optimal*, for the cases of Wiser with an honest downstream, Wiser with a dishonest downstream, and routing without normalization constraint. It shows that the traffic performance does not suffer much with Wiser but is poor when there is no constraint on cheating. Thus, even if a few ISPs continue to cheat in spite of the limited gain, the constraints imposed by Wiser ensure that traffic performance will not suffer for the topology, workloads and strategy that I study.

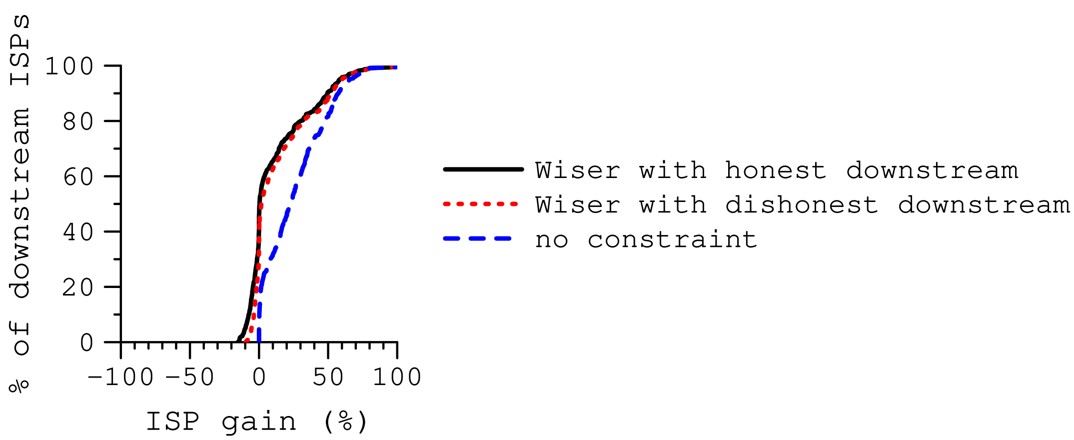
Outgoing traffic

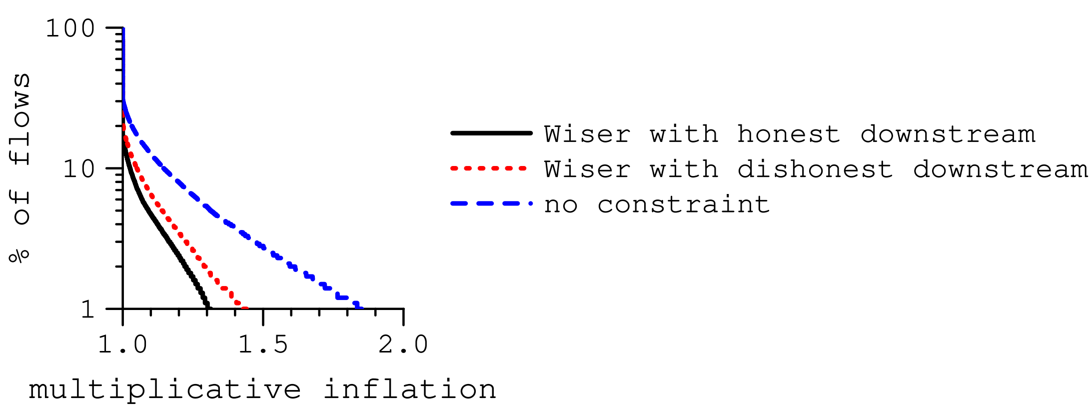
I now evaluate the robustness of Wiser to a cheating upstream ISP that attempts to reduce the cost of its outgoing traffic by dishonestly selecting paths. The incentive to pick low cost paths in Wiser is predicated on that an upstream ISP will have to make a higher virtual payment when it selects paths dishonestly than when it selects paths honestly. A higher payment will increase the payment-to-cost ratio, possibly leading to monetary penalties. Figure 5.22 plots the ratio of the average virtual payment made by the upstream ISP to the average cost announced by the downstream ISP. With Wiser, the ratio is less than 0.8 for 75% of the cases. Higher ratios with Wiser correspond to pairs of ISPs with too few routing path choices, such that *optimal*, Wiser and *anarchy* lead to similar paths. The "no constraint" curve shows the payment ratios that emerge with completely dishonest path selection. The ratios hover around 1, as expected, and 80% of the pairs have a ratio of more than 0.8.

To understand the impact of an ISP cheating within the limits of Wiser, I assume that ISPs have a contractual obligation to maintain a ratio less than 0.8. Otherwise, there are

Figure 5.21: The impact of dishonest cost disclosure on ISP gain and path length. The ISP gain is measured as the reduction in average distance relative to *anarchy*. The inflation is measured as the increase in length relative to *optimal*. *Top:* The CDF of gain for honest, upstream ISPs. *Middle:* The CDF of gain for dishonest, downstream ISPs. *Bottom:* The CCDF of inflation in path length.


% of upstream ISPs
100
80
60
40
20
0
−100 −50 0 50 100
ISP gain (%)
Wiser with honest downstream
Wiser with dishonest downstream
no constraint


% of downstream ISPs
100
80
60
40
20
0
−100 −50 0 50 100
ISP gain (%)
Wiser with honest downstream
Wiser with dishonest downstream
no constraint


% of flows
100
10
1
1.0 1.5 2.0
multiplicative inflation
Wiser with honest downstream
Wiser with dishonest downstream
no constraint

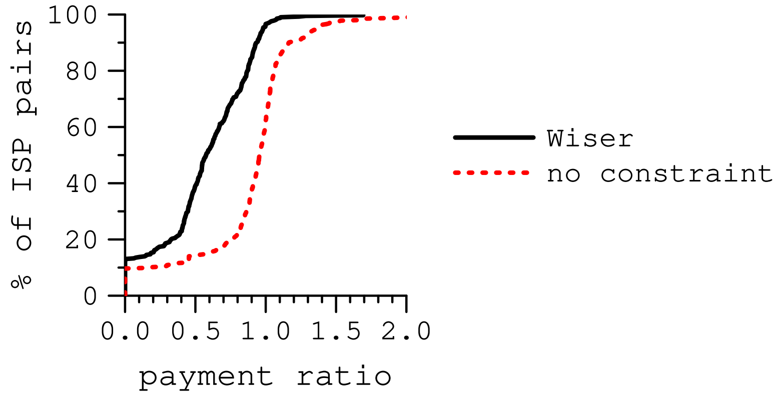


Figure 5.22: The CDFs of payment ratio with Wiser and in a scenario where the upstream ISP selects locally optimal paths.

monetary penalties or the other ISP terminates the coordination agreement altogether. This threshold is chosen for illustrative purposes; different pairs of ISPs will use different thresholds based on their situation. ISP pairs that have a ratio higher than 0.8 even with honest path selection will have a different bound and I ignore them for this analysis. (While Wiser does not directly enable ISPs to select a ratio threshold, reasonable thresholds will emerge as ISPs gain more experience with Wiser. This is similar to the emergence of the current bounds on the ratio of the traffic exchange between peers.)

Since some ISPs have a lower ratio than 0.8 when they select paths honestly, they can try to cheat by modifying their path selection such that the ratio is close to 0.8. In my simulation, an upstream ISP increases its ratio by artificially reducing the normalization factor. This reduces the relative weight given to the downstream ISP's costs, which makes the path selection more favorable to the upstream ISP. I use binary search to compute the minimum normalization factor that keeps the ratio less than 0.8. While other strategies to increase the ratio are possible, compared to the strategy that I simulate, they can only increase the gain for the upstream ISP without increasing the cost to the downstream ISP. Thus, the simulated strategy represents an upper bound on the loss for the downstream ISP.

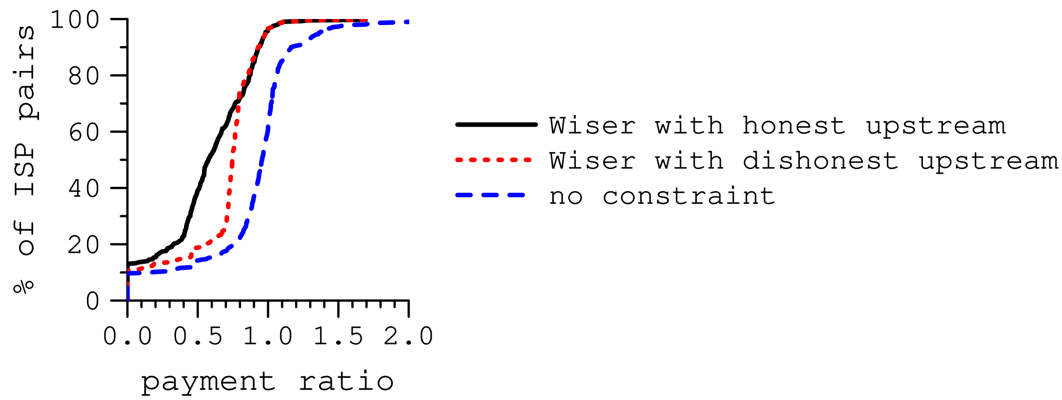


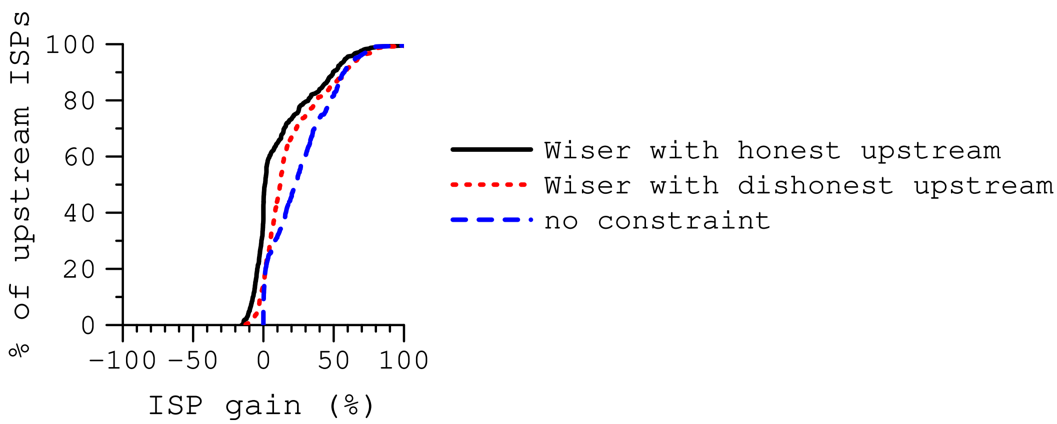
Figure 5.23: The CDFs of payment ratio with Wiser, both when the upstream ISP is honest and when it is dishonest within the bounds of Wiser, and in a scenario where the cheating is not constrained.

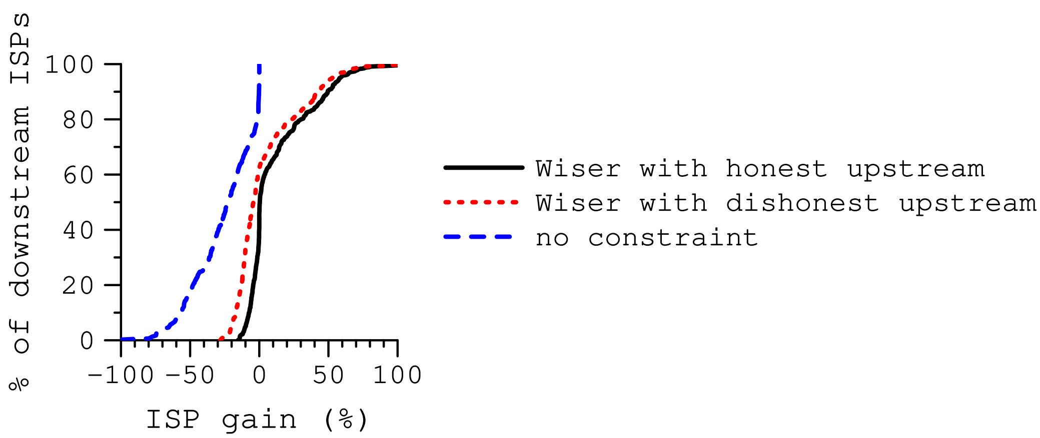
It also represents a lower bound on the traffic performance because increasing upstream gain without increasing downstream loss improves overall performance when the ISPs' metrics are similar.

Figure 5.23 shows the resulting ratio profile. As expected, many ISPs have a payment ratio around 0.8. The two previous curves for Wiser with honest upstream ISP and *no constraints* are reproduced for comparison.

Figure 5.24 shows the impact of this cheating strategy on ISP gain and path length. The top two graphs show that, for the topologies, workloads and strategy that I study, Wiser limits the gain for the dishonest ISP and the loss for the honest ISP. The respective gain and loss is high in the *no constraint* case, which implies that the payment ratio constraint helps to make cheating less effective in Wiser. The bottom graph shows that the traffic performance does not suffer much even if a few ISPs continue to cheat within the constraint imposed by Wiser.

Figure 5.24: The impact of dishonest path selection on ISP gain and path length. The ISP gain is measured as the reduction in average distance relative to *anarchy*. The inflation is measured as the increase in length relative to *optimal*. *Top:* The CDF of gain for dishonest, upstream ISPs. *Middle:* The CDF of gain for honest, downstream ISPs. *Bottom:* The CCDF of inflation in path length.


% of upstream ISPs
100
80
60
40
20
0
−100 −50 0 50 100
ISP gain (%)
Wiser with honest upstream
Wiser with dishonest upstream
no constraint


% of downstream ISPs
100
80
60
40
20
0
−100 −50 0 50 100
ISP gain (%)
Wiser with honest upstream
Wiser with dishonest upstream
no constraint

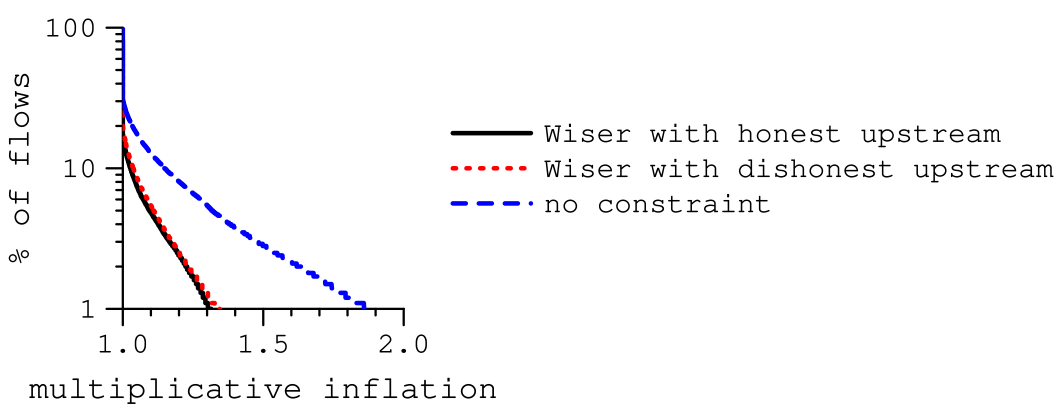
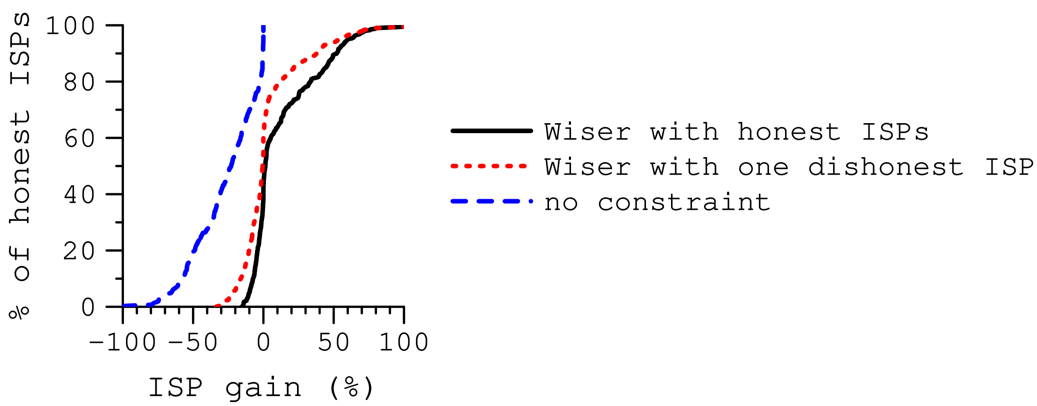
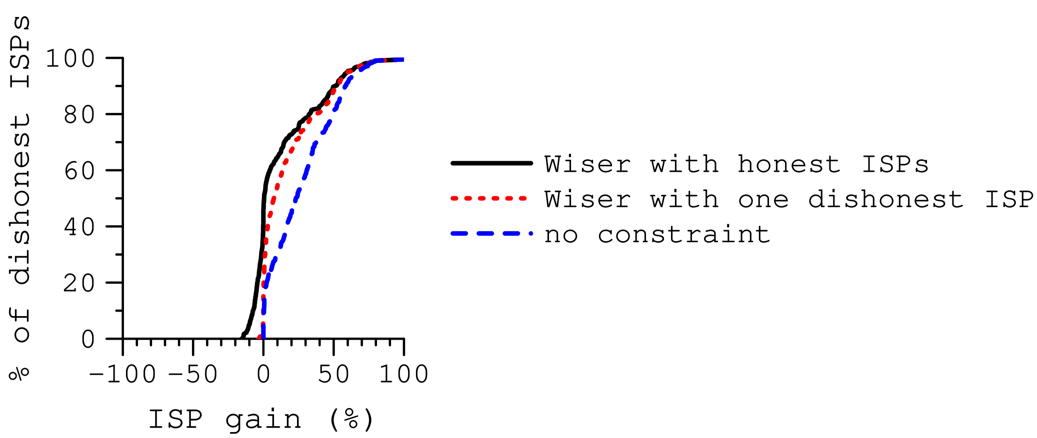
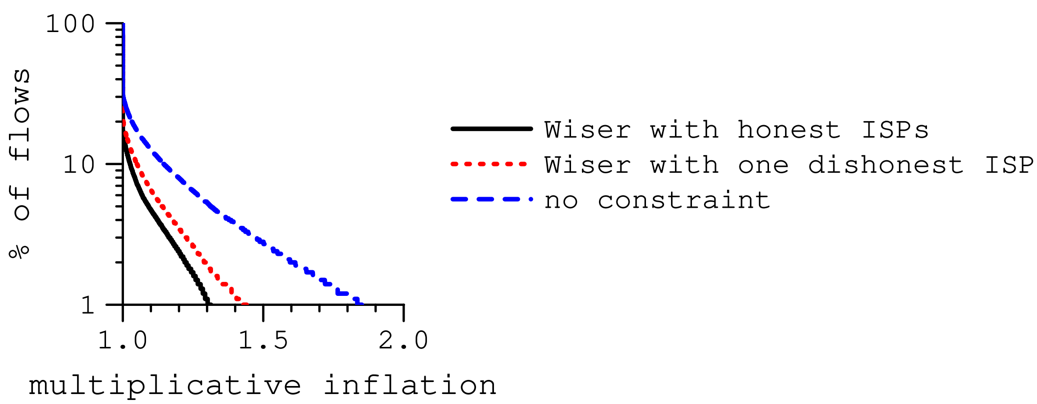

% of flows
100
10
1
1.0 1.5 2.0
multiplicative inflation
Wiser with honest upstream
Wiser with dishonest upstream
no constraint

Figure 5.25: The impact of dishonest cost disclosure and dishonest path selection on ISP gain and path length. The ISP gain is measured as the reduction in average distance relative to *anarchy*. The inflation is measured as the increase in length relative to *optimal*. *Top:* The CDF of gain for dishonest ISPs. *Middle:* The CDF of gain for dishonest ISPs. *Bottom:* The CCDF of inflation in path length.


% of honest ISPs
100
80
60
40
20
0
−100 −50 0 50 100
ISP gain (%)
Wiser with honest ISPs
Wiser with one dishonest ISP
no constraint


% of dishonest ISPs
100
80
60
40
20
0
−100 −50 0 50 100
ISP gain (%)
Wiser with honest ISPs
Wiser with one dishonest ISP
no constraint


% of flows
100
10
1
1.0 1.5 2.0
multiplicative inflation
Wiser with honest ISPs
Wiser with one dishonest ISP
no constraint

Incoming and outgoing traffic

To conclude the investigation into the robustness of Wiser to a cheating ISP that tries to reduce the average cost of traffic it carries, I evaluate the impact of a cheating ISP that uses both of the strategies above simultaneously. Figure 5.25 shows the results in the same format as before. As is the case for using these strategies individually, for the topologies, workloads and strategies that I study, the constraints imposed by Wiser limit the gain for the dishonest ISP and the loss for the honest ISP. The constraints also ensure the traffic performance does not suffer significantly.

5.5.2 Attracting traffic

In this section, I consider another motivation for an ISP to cheat. Here, a transit ISP tries to attract more traffic to its network, and thus away from other transit ISPs, so that it can collect more revenue from its customers.

I simulate a situation where a transit ISP tries to attract more traffic that originates from or is destined for its multi-homed customers, i.e., customer ISPs that connect to multiple provider ISPs. The issue of attracting traffic corresponding to singly-homed customers is moot since that traffic already traverses the ISP's network. The transit ISP cheats by hiding its internal cost for traffic corresponding to multi-homed customers, by pretending that the cost of the internal path taken by such traffic is zero. The result is that the neighbors of this ISP will be more likely to choose paths that traverse its network.

I conduct experiments corresponding to different tier-1 ISPs acting as the cheater; exactly one tier-1 ISP cheats in each experiment. The complete internetwork topology is used for these experiments. Traffic consists of a flow between each pair of PoPs.

Figure 5.26 shows the results of this experiment. The top graph plots the CDF of the percentage of additional flows that a tier-1 ISP can attract to its network using the above strategy. The cheating ISPs can attract from 3 to 27% more flows to their networks, with the average being 12%. However, while successful at attracting traffic, this behavior is

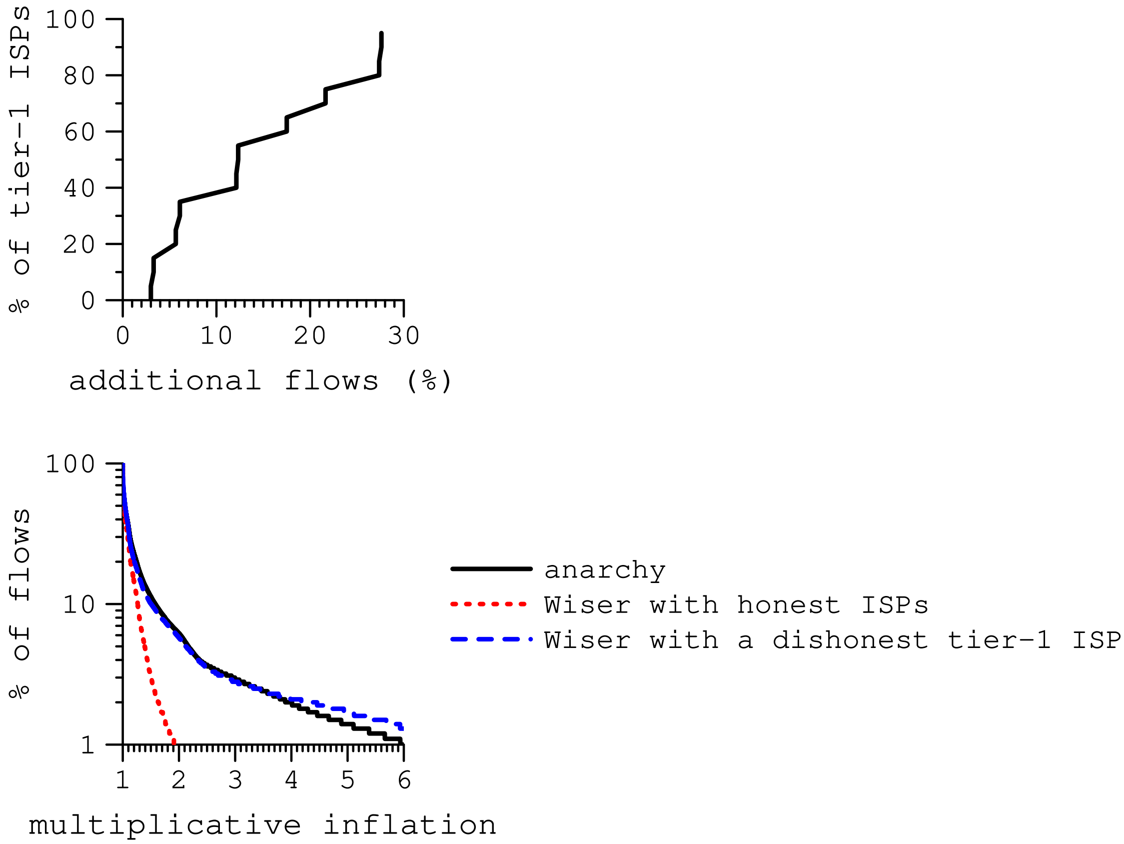


Figure 5.26: The impact of hiding internal costs on incoming traffic and path length. *Top:* The CDF of additional flows a tier-1 ISP attracts by hiding internal costs. *Bottom:* The CCDF of inflation in path length, measured as the increase relative to *optimal*.

undesirable if the ISP wants to provide good performance to its customers. The bottom graph plots the inflation in path length, relative to *optimal*, for flows that have been attracted to their networks by the cheating ISPs. For comparison, it also plots the inflation in path length with *anarchy* and with Wiser when the tier-1 ISPs act honestly. The graph shows that with cheating the path length suffers for many flows. This acts as a disincentive against such behavior because it is in the interest of the transit ISP to not indulge in behavior that leads to poorer performance for customer traffic (and it is also in the interest of the customers to monitor for this behavior).

5.6 Summary

In this chapter, I used measured ISP topologies and two independent implementations to evaluate Wiser for its efficiency, overhead and robustness to cheating. Overall, I found that Wiser leads to efficient routing with low overhead and limits the gain from cheating for the topologies, workloads and strategies that I studied.

For the scenarios where ISPs have comparable metrics and cooperate honestly, I evaluated the efficiency of Wiser for two metrics of interest to users and ISPs. For the path length metric, I found that compared to *optimal*, the average path length is only 4% higher with Wiser as opposed to being 13% higher with *anarchy*. While this may represent a useful gain in efficiency, the primary difference between the two routing methods is in the tail of the path length distribution. The top 1% of the paths are longer by a factor of six with *anarchy* and only by a factor of 1.5 with Wiser. This suggests that Wiser can be effective at optimizing the poor tail of Internet paths, a task that requires costly and unreliable manual intervention today. My results also suggest that protocols based on multilateral barter or global currency are unwarranted in the Internet, because the efficiency of bilateral barter in Wiser comes close to socially optimal routing. (In the next chapter, I posit explanations for this effect.) For the bandwidth metric, ISPs need to provision much less with Wiser than with *anarchy* for the models that I considered. The average difference is 8% which translates into significant monetary savings for ISPs if they need to upgrade their networks less frequently.

For the scenario where ISPs have diverse objectives and are cooperative, I showed that Wiser enables them to cooperate such that each gains by its own reckoning, and the efficiency comes close to being Pareto-optimal. Additionally, end-to-end performance does not suffer when ISP objectives are partially grounded in metrics of interest to end users.

I also used two independent implementations of Wiser, one in SSFNet and another in XORP, to quantify its overhead relative to BGP for several metrics of interest. I found that Wiser is easy to implement. On top of existing BGP implementations, it required fewer than

6% additional lines of code on each platform. I found that the routing message processing overhead that Wiser imposes on routers is similar to BGP. With workloads seen by routers today, the computation overhead of Wiser is within 15-25% of BGP. I also showed that Wiser has acceptable convergence time even in response to major routing perturbations.

Finally, for the topologies, workloads and strategies that I considered, I showed that the constraints in Wiser that stem from cost normalization and virtual payment ratio limit the gains for cheating ISPs and losses for honest ISPs.

Chapter 6

EVALUATION II: UNDERSTANDING THE DESIGN SPACE

In this chapter, I use a mix of analytical and empirical evaluation to explore the design space of coordination protocols for Internet routing. My goal is to understand the characteristics of the Internet topology that are responsible for the efficiency of Wiser and whether potentially simpler approaches can lead to similar efficiency. I divide this exploration into two parts.

1. Can potentially simpler approaches lead to equally efficient routing? The empirical results in the last chapter suggest that approaches that are more complicated than Wiser, such as those based on multilateral barter, are unwarranted in the Internet. A related question of interest is whether simpler protocols can be equally efficient. I explore this question in Section 6.1 in the context of the two key elements of my approach, holistic barter and agnostic costs.

- *Holistic barter* Wiser takes a holistic view of the traffic exchanged between two ISPs. While the other extreme of considering only one flow at a time is inefficient because it leads to anarchic routing, a natural intermediate point is considering pairs of flows going in opposite directions. Barter based on pairs of flows might lead to efficient routing; for instance, it would be successful in the scenario in Figure 4.1 using a simpler protocol. I show that routing based on flow-pair barter is not efficient for the topologies and workloads that I study.

- *Agnostic costs based on cardinal preferences* I investigate whether less information can be disclosed than the cardinal preferences used in Wiser, without sacrificing efficiency. While I leave a complete answer for future work, to gain insight into the matter, I compare Wiser with ordinal preferences which represent another natural point in the information disclosure spectrum. Ordinal preferences are used today with MEDs. While they disclose less information, I show that they are not as efficient as cardinal preferences for the topologies and workloads that I study.

2. Why does the bilateral barter of Wiser lead to efficient routing in the Internet? The previous chapter also showed that, for the topologies, workloads and ISP behaviors that I studied, the efficiency of Wiser comes close to that of optimal routing that optimizes the internetwork using global information. This is surprising because, in general, bilateral barter is expected to be less efficient. This suggests that certain aspects of the Internet topology may be responsible for the efficiency of Wiser. I use analysis in Section 6.2 to show how this behavior potentially results from a similarity in ISPs' cost.

6.1 Efficiency of Simpler Approaches

In this section, I investigate whether simpler approaches than Wiser might be effective at computing efficient routing paths. I conduct this investigation for two key aspects of Wiser: holistic barter and cardinal preferences. Below, I study alternatives to these and empirically evaluate their efficiency.

The methodology in this section is similar to the path length experiments in Section 5.2.1, except that here I consider only pairs of adjacent ISPs that interconnect in two or more places (so that there are multiple interconnections to choose from while routing). This allows me to focus on the efficiency of the underlying bilateral barter between adjacent ISPs. Both ISPs optimize for distance. Traffic consists of unit flows from each PoP in one ISP to each PoP in the second ISP.

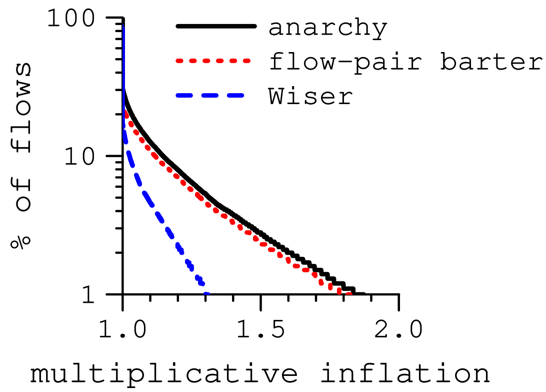


Figure 6.1: The CCDF of multiplicative inflation in path length, relative to *optimal*, with *anarchy*, flow-pair barter, and Wiser.

6.1.1 Flow-pair barter

I first evaluate whether the holistic barter is important by comparing its routing efficiency to an approach based on flow-pair barter. In flow-pair barter, ISPs consider pairs of flows going in opposite directions between two PoPs. Considering flow-pairs is a natural intermediate point between optimizing individual unidirectional flows, as is done today, and all flows in both directions, as is done in Wiser. For instance, this routing methodology would be successful in the scenario in Figure 4.1.

Flow-pair barter implements win-win routing for each pair. ISPs look for pairs of interconnections that, compared to the anarchic choice of interconnections for the two flows, lead to a gain for at least one ISP while ensuring that the other does not lose. When multiple such interconnection pairs exist, the one that minimizes the sum of the end-to-end lengths of the two paths is used. When no such interconnection pair exists, the two flows continue to use anarchic interconnections, ensuring that neither ISP loses.

Figure 6.1 plots the CCDF (complimentary cumulative distribution function) of multiplicative inflation of path length with *anarchy*, flow-pair barter, and Wiser relative to *optimal* path length. Each point corresponds to a flow across a pair of adjacent ISPs. The graph shows that, for the topologies and workloads that I study, flow-pair barter has poor efficiency; in fact, it does not lead to much efficiency improvement over *anarchy*.

6.1.2 Ordinal preferences

I now compare the efficiency of ordinal preferences to that of cardinal preferences. Ordinal preferences disclose less information and are particularly interesting because they are already disclosed by ISPs that use MEDs. While ISPs today often use their IGP costs, which are cardinal, as the basis for MEDs [75, 76], BGP considers only their relative ordering and ignores their magnitudes. The efficiency comparison helps to ascertain whether ISPs must disclose cardinal preferences to achieve efficient routing.

I compute the routing with ordinal preferences in a manner similar to that of Wiser. Downstream ISPs disclose their ordinal preference for each interconnection in the integral range $[1..N]$, where N is the number of interconnections between the two ISPs, and a lower preference corresponds to a more preferred interconnection. Upstream ISPs select the interconnection that minimizes the sum of local and remote preferences. If multiple such interconnections exist, the one with the minimum preference for the upstream ISP is chosen. The extent of information disclosed by ISPs, but not the path selection method, is thus similar to that of MEDs. I obtained similar results with another method for path selection: upstream ISPs select the interconnection that minimizes the maximum preference across the two ISPs. If multiple such interconnections exist, the one with the minimum sum of preferences is chosen.

Figure 6.2 shows the CCDF of multiplication inflation of path length, relative to *optimal*, with Wiser, ordinal preferences, and *anarchy*. Because the efficiency of early-exit routing is similar to that of late-exit routing achieved with MEDs, the curve for *anarchy* also represents the efficiency of routing with MEDs as used today. The graph shows that, for the topologies and workloads that I study, the routing efficiency achieved using ordinal preferences is not much better than *anarchy*. A closer investigation suggests that Wiser is more efficient because cardinal preferences can identify interconnections that, compared to

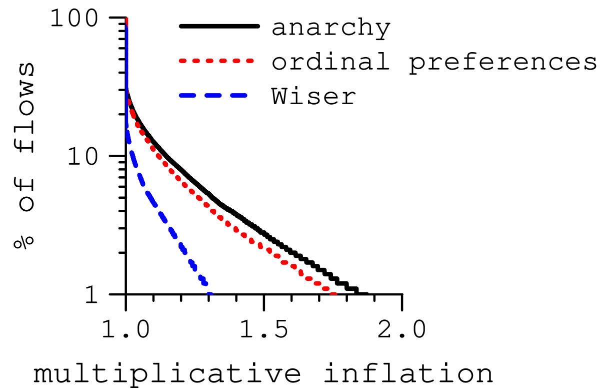


Figure 6.2: The CCDF of multiplicative inflation in path length, relative to *optimal*, with *anarchy*, ordinal preferences, and Wiser.

another interconnection, lead to a small loss for one ISP but a bigger gain for the other such that the overall path is better. Ordinal preferences are not able to do this because they do not capture the magnitude of gain or loss.

6.2 Explaining the Efficiency of Wiser

In this section, I use analytic models to gain insight into the quality of routing produced by Wiser as a function of ISP topologies. A high-fidelity model of the Internet topology depends not only on how various ISPs connect to each other but also on the internal topologies of ISPs, which are different for different ISPs [110]. Instead of constructing such a detailed model, I work with a simpler abstraction. Even this simple model, which I validate using controlled experiments with measured ISP topologies, is more realistic than the only other model of selfish ISP routing of which I am aware [54]. Combined with the empirical results presented in Chapter 5, the model helps to understand the characteristics of the Internet topology that make Wiser-like bartering successful. I first model the two ISP case and then build on that model to understand routing efficiency for the general case of multiple ISPs.

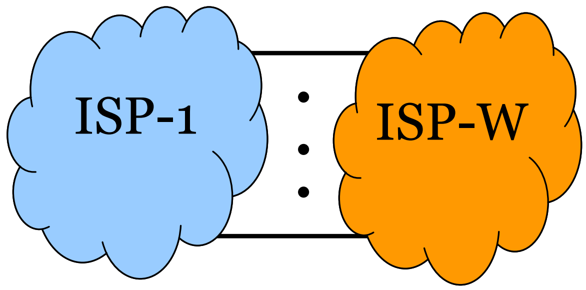


Figure 6.3: A simple analytic model of barter between two ISPs. The two ISPs interconnect in N locations. The internal ISP topology is modeled as a random mesh. The internal cost of carrying a packet between two nodes is uniformly distributed in the range [0..1] for ISP-1 and [0..W] for ISP-W.

6.2.1 Two ISP Case

Figure 6.3 shows my model for the case of two ISPs. These two ISPs, called ISP-1 and ISP-W, interconnect in N places. I model the internal ISP topology as a mesh in which the cost of transporting a packet between two nodes is drawn from a uniform random distribution in the range [0..1] for ISP-1 and [0..W] for ISP-W ($W \geq 1$). W captures the heterogeneity between the two ISPs: a higher W implies that the average cost of carrying packets inside ISP-W is higher, due to either its bigger geographic spread or other topological factors. I consider the cost of transporting a packet across the two ISPs as the sum of the costs it incurs inside each ISP. This assumes that either both ISPs have comparable objectives or that their objectives have been mapped into comparable costs.

While I focus on uniform continuous costs, this framework can accommodate other cost models. For instance, I also analyzed a bimodal cost function in which the costs for ISP-1 are 0 or 1 and for ISP-R are 0 or W. This is a simple abstraction for a congestion-based cost metric that classifies paths between two nodes as being either congested or not congested. The qualitative results with this cost function are similar to those presented below.

Realistically, the cost of the path between a pair of nodes inside an ISP network is not independent of the costs of paths between other pairs because of topological constraints.

For instance, paths between pairs of nodes that are not directly connected depend on other paths. However, this simplification makes it possible to analytically compute the costs of various routing methods, and as I confirm through controlled experiments with measured topologies, it does not detract from the main purpose of the model which is to provide insight into the efficiency of Wiser.

This model is more realistic than the earlier model proposed by Johari and Tsitsiklis [54] because it enables a characterization of the impact of two important topological factors, N and W. Their model ignores these factors as it assumes that ISP topologies are identical (W=1) and computes the worst-case inefficiency independent of the number of interconnections. I show that efficiency of routing paths depends on these factors.

I now use my model to compute the expected cost of routing using different methods and quantify their relative efficiencies.

Routing Costs

The goal of interdomain routing in the two ISP case is to select interconnections for traffic that crosses ISP boundaries. Consider the following four routing methods for selecting interconnections.

1. **Early-exit routing (or *anarchy*)** selects the interconnection that is closest to the source of the packet inside the upstream ISP. Consider a packet going from ISP-1 to ISP-W. The expected cost of transporting this packet inside the upstream ISP is the expected value of the minimum of N random numbers. These numbers represent the costs from the source to the interconnection points, of which the upstream ISP picks the minimum. This cost is $\int_0^1 x\ N\ dx\ (1-x)^{N-1} = \frac{1}{N+1}$; the probability of x being the minimum out of N numbers is $N\ dx\ (1-x)^{N-1}$, and integrating it after multiplying by x yields the expected value of the minimum. The expected cost inside the downstream ISP is $W/2$ because that is the cost to carry a packet from an interconnection to a randomly selected destination. Adding the costs of both ISPs yields $\frac{1}{N+1} + \frac{W}{2}$, which is the total expected cost of a packet

going from ISP-1 to ISP-W. Similarly, the expected cost of a packet going in the other direction is $\frac{1}{2} + \frac{W}{N+1}$.

Assuming that the traffic in the two directions is equal, the expected cost of all traffic is $C_{early}(N,W) = \frac{(W+1)(3+N)}{4(N+1)}$. The individual costs to the two ISPs are $C^1_{early}(N,W) = \frac{N+3}{4(N+1)}$ and $C^W_{early}(N,W) = \frac{W(N+3)}{4(N+1)}$.

2. Optimal routing selects the interconnection that minimizes the total cost of the packet across the two ISPs. The cost of a packet through an interconnection can be modeled as the sum of two random numbers and the expected cost of optimal routing is the expected value of the minimum sum. I derive the expected cost of optimal routing in Appendix A and present only the final result below.

The expected total cost of optimal routing is:

$$\begin{aligned}
C_{optimal}(N,W) &= \frac{N}{3W}\,{}_2F1(\frac{3}{2}, 1-N, \frac{5}{2}, \frac{1}{2W}) \\
&+ \frac{N}{2(2W)^N}\left(\frac{(2W+1)((2W-1)^N-1)}{N} - \frac{(2W-1)^{N+1}-1}{N+1}\right) \\
&+ \frac{2WN+W+1}{(2N+1)(2W)^N}
\end{aligned}$$

where ${}_2F1$ is the Gauss hyper-geometric function: ${}_2F1(a,b,c,z) = \sum_{k=0}^{\infty} \frac{\binom{a}{k}\binom{b}{k}z^k}{\binom{c}{k}k!}$.

The individual cost for ISP-1 is:

$$\begin{aligned}
C^1_{optimal}(N,W) &= \frac{N}{6W}\,{}_2F1\left(\frac{3}{2}, 1-N, \frac{5}{2}, \frac{1}{2W}\right) \\
&+ \frac{(2W-1)^N-1}{2(2W)^N} \\
&+ \frac{N+1}{(2N+1)(2W)^N}
\end{aligned}$$

And for ISP-W is:

$$C^W_{optimal}(N,W) = C_{optimal}(N,W) - C^1_{optimal}(N,W)$$

3. Wiser selects an interconnection that minimizes the cost after normalization. The costs of ISP-W will be normalized by dividing by W. At this point, the normalized cost of routing with Wiser is same as that of optimal routing with $W = 1$. Thus, individual costs for ISP-1 is $C^1_{Wiser}(N, W) = C_{optimal}(N, 1)/2$ and for ISP-W is $C^W_{Wiser}(N, W) = \frac{WC_{optimal}(1)}{2}$. The expected total cost is $C_{Wiser}(N, W) = \frac{(W+1)C_{optimal}(1)}{2}$.

4. Random-exit routing selects an interconnection for each flow randomly. This method is not prevalent in the Internet; I analyze its efficiency only as a point of comparison. Its expected cost is $C_{random}(N, W) = \frac{W+1}{2}$. The expected costs of routing within ISP-1 is $C^1_{random}(N, W) = \frac{1}{2}$ and within ISP-W is $C^W_{random}(N, W) = \frac{W}{2}$.

Relative Efficiency

I now study the efficiency of the different routing methods as a function of W, the topological factor that captures the heterogeneity between ISP costs. I use cost inflation, which is the ratio of the cost of the given routing method and the cost of *optimal*, as the measure of efficiency. This measure captures the average inflation, not the worst-case inflation, for individual flows. This metric underestimates the benefit of Wiser in that it discounts the impact of egregiously bad cases that operators need to manually improve today. To confirm the existence of such cases, I use simulation to study the distribution of flow costs.

I empirically validate model predictions using measured ISP topologies. Details on measured topologies are presented in Chapter 5. To study the impact of W in isolation, I take a measured ISP topology, corresponding to ISP-1, and assume that it interconnects with a replica of itself, corresponding to ISP-W. To mimic today's networks in which interconnections are usually present in well-connected cities, only nodes that connect to at least three other nodes are chosen (randomly) to be interconnections. The costs inside ISP-1 are based on link length, and to simulate different values of W, the costs inside ISP-W are scaled accordingly. Traffic over this two ISP topology consists of unit flows from each node inside one ISP to each node inside the other ISP.

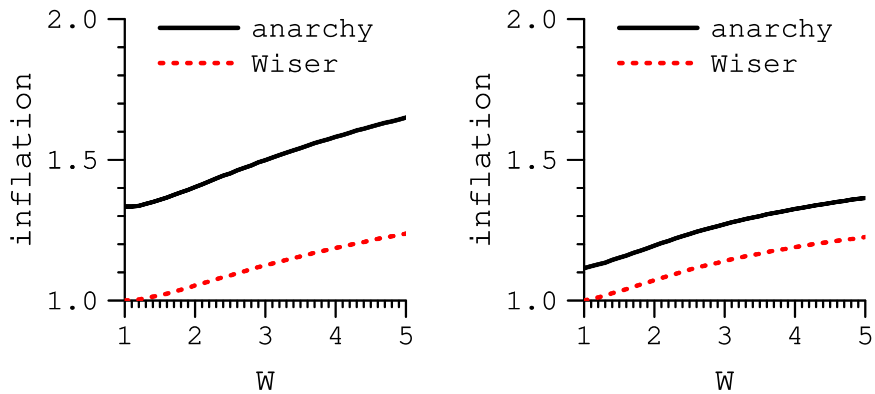


Figure 6.4: Cost inflation with *anarchy* and Wiser relative to *optimal*. *Left:* Analytical results. *Right:* Results using the AT&T topology.

Figure 6.4 plots the inflation due to *anarchy* and Wiser as a function of W. The number of interconnections, N, is set to six. The left graph shows the average inflation predicted by the model. Wiser is always more efficient than *anarchy*. It closely approximates *optimal* for low values of W but is less efficient for higher values of W; I explain this effect below. The right graph shows the experimental results using AT&T, a tier-1 ISP, as the input topology. Qualitatively similar results are obtained with other tier-1 ISPs. The trends largely agree with the model though the inflation with *anarchy* is lower, likely due to simplifying assumptions in the model, such as the independence of costs of paths between two nodes.

Although not shown in the figure, *anarchy* is much better than random-exit routing. For instance, the inflation at $W = 5$ with random-exit routing is roughly 3, while the inflation with *anarchy* is only 1.8 (left graph). Thus, the act of each ISP doing the best for itself, even without regard for others, leads to a more efficient system compared to completely oblivious routing.

Figure 6.5 shows that the higher average inflation of *anarchy* leads to a poor tail for both the model and the AT&T topology. The model results are obtained by simulating one thousand flows from ISP-1 to ISP-W and an identical number of flows in the opposite

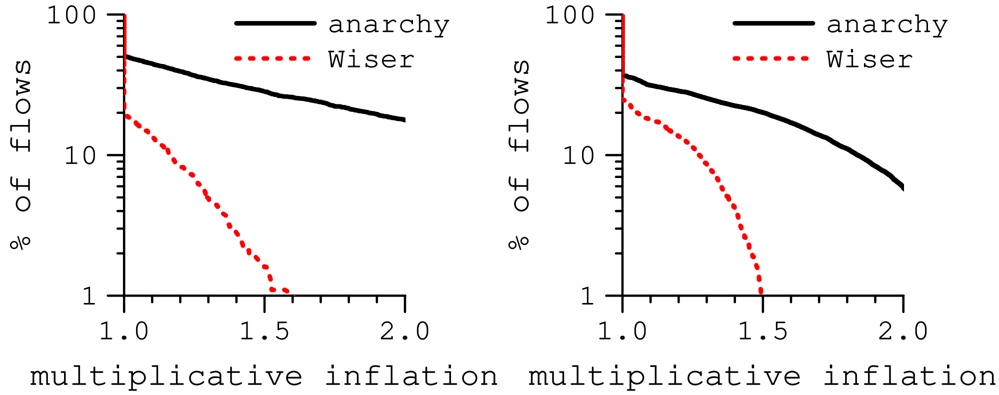


Figure 6.5: The CCDF of multiplicative cost inflation with *anarchy* and Wiser relative to *optimal*. The value of W is set to two. *Left:* Results using the model. *Right:* Results using the AT&T topology.

direction. Traffic for the AT&T topology is the same as above. The graphs plot the CCDF of multiplicative inflation in flow costs relative to *optimal*. As is observed in practice, they show that while most flows are not inflated, some of them are significantly inflated with *anarchy*.

To investigate why Wiser is less efficient than *optimal* for higher values of W, Figure 6.6 plots the gain of Wiser and *optimal* for individual ISPs. Gain is computed as the reduction in cost compared to the cost of *anarchy*. The left graph shows the analytical results and the right graph shows the results using the AT&T topology. The graphs show that while with Wiser both ISPs have similar relative gains, with *optimal*, ISP-W gains at the expense of ISP-1. Thus, because *optimal* disregards ISP boundaries, ISP-1 suffers for the greater good. In Wiser, the win-win constraint reduces overall efficiency when the ISPs are very diverse. That the efficiency of Wiser comes close to that of optimal routing for the topologies studied in Chapter 5 suggests that the costs of ISPs that interconnect in multiple places are roughly similar, at least for the metrics that I studied.

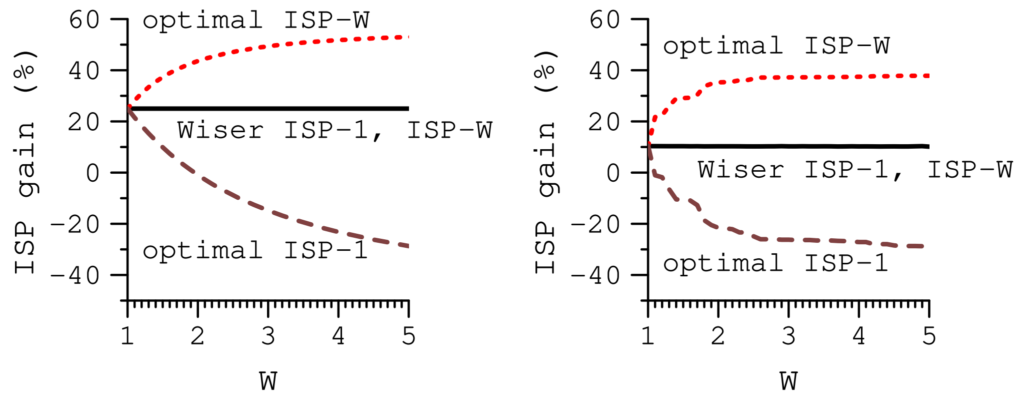


Figure 6.6: Gain for individual ISPs with Wiser and *optimal*, measured as the reduction in cost compared to *anarchy*. *Left:* Analytical results. *Right:* Results using the AT&T topology.

6.2.2 Multi-ISP Case

I now try to understand the efficiency of *anarchy* and Wiser in the general case of multiple ISPs. Unlike the previous section, where I used analysis to compute the average cost inflations of these methods, I use simple arguments to provide insight into their worst case inflations relative to *optimal*.

Consider two nodes in the Internet. Assume that the *optimal* path P_0 between these nodes has cost C_0 and consists of k_0 ISPs. $C_0 = c_0^{i_0^1} + c_0^{i_0^2} + \cdots + c_0^{i_0^{k_0}}$, where i_0^1 is the source ISP, $i_0^{k_0}$ is the destination ISP, and $c_0^{i_0^j}$ is the cost inside the j-th ISP. Consider another potential path P_1 between the two nodes with cost C_1 ($\geq C_0$) and with k_1 ISPs along the path. In similar notation, $C_1 = c_1^{i_1^1} + c_1^{i_1^2} + \cdots + c_1^{i_1^{k_1}}$.

Now consider the path selection of the two routing methods. *Anarchy* selects paths based purely on the internal cost of the source ISP and can select P_1 independently of how much higher C_1 is compared to C_0. Thus, the worst case inflation with *anarchy*, relative to *optimal*, is bounded only by the ratio of the costs of the costliest path and the optimal path.

With Wiser, after a series of cost normalizations, the costs of the two paths as perceived by the source ISP will depend on the normalization factors of the ISPs along the path. For instance, for P_0, the perceived cost will be $c_0^{i_0^1} + w_{i_0^2}^{i_0^1}(c_0^{i_0^1} + w_{i_0^3}^{i_0^2}(\cdots)))$, where w_b^a is the normalization factor between ISP-a and ISP-b. The costlier path, P_1, will be chosen only when the normalization factors are such that its perceived cost is less than that of P_0. Thus, the worst case inflation with Wiser is a (complex) function of normalization factors in the topology. The inflation will be lower when the normalization factors in the network are closer to one. Additionally, because the nature of Internet routing is such that packets spend most of their time inside one or two large ISPs that sell transit to other ISPs [45], the normalization factors between these transit ISPs have the most impact on path selection. If the normalization factors between these ISPs is close to one, the inflation due to Wiser will be low. The last chapter shows that the efficiency of Wiser is close to that of *optimal*, which suggests competing transit ISPs have similar costs for the topologies and ISP objectives that I study.

6.3 Summary

In this chapter, I used empirical evaluation to understand whether approaches that are potentially simpler than Wiser can produce efficient routing, and I used analysis to understand the characteristics of the Internet topology that help Wiser find efficient routing paths. For the ISP topologies that I considered, I showed that holistic barter is key to efficiency with win-win routing; barter over pairs of flows, a natural intermediate point between considering individual flows and all traffic, does not result in efficient routing. I also showed that ordinal preferences that ISPs disclose today as part of MEDs do to result in efficient routing. My model predicts that the efficiency of routing with Wiser is high when ISPs, especially large transit ISPs, have similar costs.

Chapter 7
RELATED WORK

In this chapter, I place my work in the context of existing research. I identify three themes of related research. The first theme focuses on optimizing interdomain traffic; the second theme examines the inefficiency of routing with competing interests; and the third theme is protocol design in competitive environments. I consider each in turn.

7.1 Optimizing Interdomain Traffic

There is a large body of work aimed at optimizing interdomain traffic, which can be classified into several distinct approaches. Table 7.1 lists these approaches and points out for each approach whether it meets the requirements mentioned in Chapter 3 and leads to efficient routing. Below, I discuss these approaches and compare them to Wiser.

7.1.1 BGP-based optimization

Perhaps the most common approach to optimizing interdomain traffic is to work within the constraints of BGP. In this approach, ISPs independently control their incoming and outgoing traffic by tuning their routing messages and path selection policies. Many examples of this approach exist, including several that are embedded in e-mail exchanges between operators, such as those on the NANOG mailing list [83]. All of them tackle a specific aspect of overall interdomain traffic optimization. MEDs and AS-path prepending which were described in Chapter 2 belong to this class of techniques. I describe a few others below and then contrast this approach with Wiser.

- Uhlig and Bonaventure address the problem of optimizing outgoing traffic from stub ISPs [119], or ISPs that do not provide transit to other ISPs. They propose that instead

Table 7.1: A comparison of various approaches for optimizing interdomain traffic. An “X” implies that the approach in the corresponding column satisfies the criterion in the corresponding row. The classification for each approach is explained in the corresponding subsection.

	BGP based	**Currency based**	**Mechanism design**	**SMPC**	**Admission control**	**Interdomain QoS**	**Overlays**	Wiser
Limits information disclosure	×			×	×		×	×
Supports diverse ISP Objectives	×	×	×		×			×
Win-win	×				×			×
Robust to cheating	×		×	×			×	×
Low overhead	×	×	×		×	×		×
Efficient		×	×	×		×	×	×

of manually tweaking outgoing traffic, as is done today, network operators specify their objective, such as an even distribution of traffic across all interconnections, to a central network controller. Based on the amount of traffic for various destinations, the controller decides via which interconnection the traffic to each destination should exit. The controller implements its decisions by sending appropriate messages to routers within the ISP.

- Quoitin *et al.* address the problem of optimizing incoming traffic to stub ISPs [91]. They propose that ISPs use *redistribution* community attributes in their routing messages. These attributes control further distribution of routing messages by upstream ISPs [19], which determines how traffic enters the ISP. For instance, if a stub ISP uses redistribution communities to request an upstream ISP to not propagate the routing messages for certain destinations to its peers, traffic from those peers will not come in through the interconnection between the stub and the upstream ISP.

  A few commercial products [52, 101] also address the problem of managing incoming and outgoing traffic for stub ISPs, but the techniques that they use are proprietary.

- Feamster *et al.* propose a set of recommendations that transit ISPs should follow while re-configuring path selection policies to control outgoing traffic [37]. Their recommendations are based on the assumption that the neighboring ISPs employ common path selection policies, so that the behavior is predictable.

  Uhlig and Quoitin also address the problem of managing outgoing traffic from transit ASes [120]. As for their work for stub ISPs [119], they develop a centralized controller that decides how traffic leaves each border router, based on the transit ISP's optimization criteria. The controller also tries to minimize the number of configuration or routing messages that need to be sent to internal routers.

The key characteristic of the BGP-based approach, and thus that of all proposals above, is that it does not require any coordination among ISPs. This is both a limitation and an advantage. The limitation is that it results in inefficient routing because of two reasons. First, in the absence of coordination, ISPs have no incentive to be sensitive to other ISPs' concerns. This leads to locally optimal routing decisions that are not as efficient from a global perspective. It can also lead to instabilities. While ISPs can minimize instabilities by opting for mechanisms that are less likely to evoke a counter-response from other ISPs, doing so limits their ability to achieve their objectives [37]. Second, even if an ISP wanted to be sensitive to other ISPs' concerns in the interest of efficiency, limited visibility into their networks acts as an impediment. In contrast, through the disclosure of agnostic costs and ISP coordination, Wiser produces efficient routing by encouraging each ISP to be sensitive to other ISPs' concerns.

The advantage of the BGP-based approach is that, in the absence of any coordination, ISPs can independently select their optimization criteria and need not worry about other ISPs manipulating the coordination mechanism. Wiser retains these advantages by using agnostic costs and limiting the loss to honest ISPs due to cheating.

The BGP-based approach satisfies all of the criteria listed in Table 7.1 except that it does not produce efficient routing. I consider it to be win-win by definition; no ISP loses compared to today's unilateral routing.

7.1.2 Currency-based routing

Another approach to interdomain traffic optimization is to use real money as the basis of path selection. Two separate proposals employ this approach [3, 79]. Both propose that downstream ISPs include the monetary price of carrying traffic as part of their routing messages. Upstream ISPs pay this price when they choose to send traffic to that destination along that route. They are expected to pick the cheapest routes, based on local costs and advertised prices. Circuitous paths would become less common because they are likely to have a higher monetary cost.

I believe that using real money at such a fine granularity is unsuitable for routing in the Internet because of several reasons. First, it assumes that ISPs are able to compute monetary costs or prices at such a fine granularity, which can be difficult if not impossible [107]. For instance, when the price of an overloaded link must include the cost of poor performance experienced by an ISP's customers, how might an ISP estimate this price? Second, it requires ISPs to disclose path prices. While prices do not directly disclose the underlying monetary costs, the ability of this approach to compute cheap paths relies on prices being closely correlated with monetary cost. ISPs are usually reluctant to share such sensitive information. Third, this approach appears incompatible with the current charging model in the Internet. It assumes a sender-pays model, but monetary payments today are independent of the direction of data transfer. Customer ISPs pay provider ISPs for both incoming and outgoing traffic.

Wiser, on the other hand, does not suffer from the limitations above. Agnostic costs are easy to compute; they can be based on easily measurable path performance metrics, as is done today for intradomain traffic optimization. They allow an ISP to control how much information it discloses. Wiser retains the current charging model of the Internet.

The currency-based approach does not satisfy three of the criteria listed in Table 7.1. It requires ISPs to disclose sensitive cost information. It is not win-win because routing using currencies could represent a higher cost for some ISPs than routing today. It is also not robust to cheating because, to maximize payment, an ISP can modify its prices based on what it observes about other ISPs' prices.

7.1.3 Mechanism design

Strategy-proof mechanisms that are provably robust to manipulation have received much attention in recent years. Distributed algorithmic mechanism design (DAMD) [40, 39] is a branch of this domain that is particularly suitable for networked systems because the mechanisms that it produces have low computational complexity and are amenable to a distributed implementation. Sami *et al.* apply DAMD to the scenario of Internet routing [38]. They propose a *direct* mechanism in which the ISPs disclose their monetary costs of carrying traffic. The disclosed costs are used to compute the cheapest routing paths and the monetary payments that source ISPs must make to the other ISPs along the path. The payments are computed based on a Vickrey-Clark-Grove (VCG) mechanism, which ensures that even if an ISP has complete knowledge of other ISPs' monetary costs, it can never lie about its own costs in a way that increases its payment.

Even though this protocol is robust to ISPs that lie about their costs, it does not capture all real-world competitive concerns [40]. For instance, an ISP can use the knowledge of the competitor's monetary costs to plan its own network in a way that undercuts the competitor's profits. In contrast, Wiser requires ISPs to disclose only agnostic costs.

In terms of meeting the criteria listed in Table 7.1, the mechanism design approach is similar to the currency-based approach except that it is robust to cheating.

7.1.4 Secure multi-party computation

A novel approach for cooperative interdomain routing without requiring ISPs to disclose their sensitive information is to use secure multi-party computation (SMPC) [47]. Using SMPC, parties can jointly compute a function that is based on inputs from all of them, without directly disclosing their inputs to others. While in theory it is possible to compute any function using this technique, in practice the computation and message overhead can be prohibitive for many problems of interest. Recently, Machiraju and Katz applied SMPC to the limited setting of routing between two ISPs [69, 68]. They enabled neighboring ISPs to select interconnections such that the maximum link utilization across the two ISPs is minimized.

While Wiser enables ISPs with diverse objectives to cooperate and focuses on win-win solutions, the work above assumes that both ISPs want to minimize maximum link utilization across both networks and that ISPs are willing to lose for the greater good. However, these limitations may not be fundamental to SMPC, and an interesting avenue for future research is investigating whether Wiser can be implemented using SMPC to further reduce the amount of information ISPs directly disclose to each other.

Because there are no complete solutions based on SMPC, my characterization of this approach in Table 7.1 is based on an extrapolation from the existing solution for the two-ISP case. I consider the SMPC approach to be robust to cheating under the assumption that ISPs will find it hard to game the system without information about other ISPs. The SMPC approach is likely to have a high overhead in terms of its routing message complexity and computational requirements.

7.1.5 Aggregate Admission Control

Winick *et al.* propose a method by which two neighboring ISPs can make interdomain routing changes cooperatively [127]. Their method is targeted for situations in which an upstream ISP might overload the downstream ISP by moving the traffic that it sends. Before moving traffic, the upstream ISP informs the downstream ISP of changes that it intends to make, and the downstream ISP decides if those changes are acceptable. This method is thus akin to admission control. Aggregation based on how various destinations attach to the downstream ISP is used to limit information disclosure and improve scalability. The authors do not consider strategic behavior by ISPs. For instance, the downstream ISP can turn down all requests except those that maximize local gain.

This work targets situations where a routing change is desired by one of the ISPs. Unlike Wiser, it does not enable continuous optimization for all traffic. Additionally, when the number of changes that are acceptable to both ISPs is small compared to the number of possible changes, many iterations would be required to discover a mutually acceptable solution.

Table 7.1 characterizes the admission control approach based on an extrapolation from the two ISP solution. I assume that it is win-win because downstream ISPs can refuse changes that cause significant losses. It is not likely to result in efficient routing because the changes proposed by the upstream ISP are based on a view of only its own network.

7.1.6 Interdomain QoS

Interdomain QoS, or quality of service, is another approach to improve the quality of Internet routing [128]. Since QoS is an overloaded term in the literature, I discuss it in the context of a commonly accepted framework [34]. In this framework, as is done in Wiser, downstream ISPs attach path quality information, such as length, to their routing messages and upstream ISPs select paths that maximize path quality. Optionally, resources are reserved along the paths that are chosen to send traffic [131]. Unlike Wiser, QoS proposals assume that all ISPs share a common metric which they are willing to disclose.

While research on QoS has been underway for more than a decade (e.g., see reference [20]), there is no significant deployment of interdomain QoS in the Internet. This might be because the interests of individual ISPs have been largely ignored by this body of work. It attempts to approximate socially optimal routing and thus suffers from the shortcomings of such routing mentioned earlier. In particular, it requires disclosure of sensitive information, requires ISPs to agree on common optimization criteria (which has proven to be a challenge for researchers that attempt to define such criteria [53]), and can lead to win-lose routing. Therefore, Table 7.1 shows interdomain QoS approaches as not satisfying any criterion except efficiency and low overhead. Because of such concerns, Wiser is designed to reflect ISP interests.

7.1.7 Overlay networks

A radically different approach to improve the end-to-end quality of Internet paths is through the use of overlays. In this approach, a group of end hosts enable applications to bypass the "default" path offered by Internet routing by relaying packets between each other. Researchers have shown that this can improve application performance [105, 5, 4].

I do not consider overlays to be a technology that directly competes with Wiser. Current overlays do not scale enough to be able to route all Internet traffic. Unlike ISPs, who compete with each other and are concerned with both end-to-end performance and internal routing costs, overlays are concerned only with end-to-end performance.

Table 7.1 characterizes the overlay based approach. Overlays do not require ISPs to disclose any information. They do not allow ISPs to optimize for their own objectives and do not aim for win-win routing; both these factors are likely to lead to a conflict between overlays and the underlying ISPs. With overlays, the issue of cheating does not arise because there is no coordination between competing parties.

An interesting avenue for future work is to revisit overlays assuming that ISPs use Wiser for interdomain routing. For instance, does the fact that Wiser produces better default routing obviate the need for overlays? Additionally, while overlays can optimize for measurable path performance metrics, they cannot optimize for certain other metrics, such as the monetary cost of a link, that may be of interest to ISPs. Can Wiser-like routing be used between ISPs and overlays to make the latter sensitive to the concerns of ISPs?

7.2 Examining the Inefficiency of Selfish Routing

In this section, I review work that empirically or analytically examines the "price of anarchy" [87] in routing in the presence of competing interests. I divide this work into two categories. The first considers the impact of selfish ISPs in the Internet, and the second considers selfish end users.

7.2.1 Selfish ISPs

My work is motivated by empirical observations made by other researchers regarding the impact of selfish ISP routing on end-to-end Internet paths. Savage *et al.* [105] first showed that a subset of "default" routing paths in the Internet have poor latency and loss rate even though better paths exist. Andersen *et al.* then built a system called Resilient Overlay Network (RON) and showed that the reliability and throughput of traffic can be improved by leveraging non-default paths [5]. While both use live measurements over a subset of

Internet paths, other work has combined measured ISP topologies with models of Internet routing to fully characterize the impact of selfish routing. This methodology is similar to that in Chapter 5. Tangmunarunkit *et al.* compute inflation in the number of router hops when comparing default paths to optimal paths [117, 118]. Spring *et al.* compute this inflation in terms of path length [109].

The conclusions of the work above concur with my results: while the inflation due to current Internet routing is small on average, it is significant for a small fraction of paths. More importantly, I also show that this inflation can be almost completely eliminated without requiring the ISPs to give up their autonomy.

The only work to my knowledge that analytically examines the impact of selfish ISP routing is that of Johari and Tsitsiklis [54]. They use a graphical argument, along with certain assumptions about ISP topologies, to show that the length of paths due to early-exit routing in a two-ISP scenario is bounded by three times the optimal path length. I present an alternate, more realistic model for this scenario and validate it using controlled experiments. My model suggests that path inflation is a function of the differences in ISP costs; it is unbounded for arbitrary networks but low when the ISP costs are similar.

7.2.2 Selfish users

Many researchers have analytically examined the price of anarchy in a network where end users route their packets selfishly. Successive works have considered an increasingly complex, and arguably realistic, network model. In their seminal work, Koutsoupias and Papadimitriou consider a simple network with two parallel links and obtain a bound on the price of anarchy, or the worst-case ratio between the latencies of selfish and optimal routing [61]. Czumaj and Vöcking extend this work to compute the worst-case ratio for any number of parallel links [35]. Roughgarden and Tardos compute a bound on the price of anarchy for general networks with an infinite number of users [100]. Recently, Roughgarden considered the case of a finite number of users, assuming that flows are fractionally routed [99], and Awerbuch *et al.* considered the case where flows are not fractionally routed [11]. Roughgarden and others have also analytically investigated the effectiveness of mechanisms that aim to reduce the price of anarchy in this setting [97, 98, 32, 33].

In contrast to the above, I consider the price of anarchy for the scenario of selfish ISPs. Unlike the above analyses which predict a worst-case bound on the price of anarchy, my analysis suggests that the price of anarchy is unbounded for arbitrary networks. This results from a fundamental difference in the two scenarios. While selfish users consider end-to-end path quality, which limits the use of egregiously bad paths, selfish ISPs in my model consider only the part of the path inside their own network. In practice, this can happen when ISPs lack information about the quality of paths inside other ISPs. As a result, routing paths can be arbitrarily inefficient.

Qiu *et al.* use measured ISP topologies to empirically investigate the price of anarchy in a network with selfish users [89]. They find that this price is low in practice because the worst-case ratios predicted above rarely arise in realistic topologies. Similar differences between realistic scenarios and theoretical bounds for the case of selfish ISPs motivated me to evaluate Wiser using realistic topologies.

7.3 Protocols in Other Environments with Competing Interests

Internet routing is only one example of a networked system with competing interests. In this section, I review protocols developed for other such environments. Because the detailed design of these protocols is largely governed by the constraints of their target environments, I compare only their high-level approaches with Wiser.

7.3.1 Multi-hop wireless networks

The role of incentives in multi-hop wireless networks that are composed of independent users has received much attention in recent years [112, 23, 133, 71, 92, 22, 134]. In such a network, end-to-end connectivity is enabled by users relaying packets for each other. But relaying consumes energy and can decrease throughput, which incents users to not relay others' packets.

Protocols that incent users to relay packets can be divided into three categories. The first category uses virtual currency [133, 23, 92]. A user is incented to relay packets because in return it earns currency which it needs to have its own packets relayed. These schemes rely on a trusted central authority to ensure the integrity of the currency. Wiser bypasses the need for a central authority by relying on barter between adjacent ISPs. This is made possible by the nature of Internet routing where adjacent ISPs are in a position to trade favors. Highly asymmetric workloads, where one node needs the neighboring node to relay packets but not the other way around, make bilateral barter less effective for multi-hop wireless networks [70].

The second category formulates relaying as a game theoretic problem and sets up the protocol such that relaying is the rational strategy [112, 134]. Such techniques rely on specific assumptions regarding workload and the objectives of individual users. In contrast, Wiser enables ISPs with different objectives to coordinate and efficiently forward packets for each other.

The final category uses enforcement to discourage nodes from shirking their relaying responsibilities [74, 22, 71]. These techniques assume that most users in the system are cooperative and instead focus on "raising the bar" to discourage the few potential cheaters. Usually, this leads to more lightweight protocols compared to the two categories above. Wiser shares this design philosophy; in the interest of efficiency and practicality, rather than making dishonest behavior impossible under all scenarios, Wiser leverages the properties of the environment to discourage it by limiting potential gains.

7.3.2 Peer-to-peer networks

Another domain where the role of incentives has received much attention in recent years is peer-to-peer networks [31, 65, 104, 2]. In such networks, users form an overlay to share resources such as media content. For such networks to succeed, it is essential that users contribute resources, instead of only consuming them.

The most successful protocol to incent users to contribute resources is BitTorrent [31]. The central incentive strategy in BitTorrent is bilateral barter, where a user is more likely to share resources with another user when the favor is reciprocated. This is of course similar to Wiser, though the mechanisms to implement barter are different in the two systems. Another aspect that is common to both protocols is favoring practicality over being strategy-proof. Determined users can cheat in BitTorrent to gain unfair advantage [108].

7.3.3 Federated distributed systems

In federated distributed systems, independent parties pool their resources to enable new services or more efficient operation. Examples include Web services [18, 58], grids [24, 43] and stream-processing [1, 12, 27]. Peer-to-peer networks are also an example of such a system with the unique property that the parties are not easily identifiable entities.

Most cooperation mechanisms designed for these systems are based on setting up virtual markets with pricing and auctions [24, 41, 43, 115, 123], which as explained earlier, are not practical in the context of Internet routing. One exception is the work of Balazinska *et al.* who tackle the problem of load management in such systems [13]. They propose a mechanism by which an overloaded party can transfer load to other parties. This work is similar to Wiser in two key respects. First, because of simplicity with respect to negotiation and enforcement, the contracts between parties are bilateral rather than multilateral, even though multilateral contracts are likely to lead to more even load distribution. Second, offline contracts are leveraged to simplify online operation, by bounding the range of the price a party needs to pay for transferring load. Wiser uses offline contracts to limit the cost an ISP incurs while carrying traffic received from a neighbor.

Chapter 8

CONCLUSIONS AND FUTURE WORK

In this chapter, I summarize my work, review its thesis and contributions, and discuss directions for future research.

In this dissertation, I presented Wiser, a practical protocol for discovering efficient Internet routes in the presence of competing ISPs. Wiser is based on bilateral barter and agnostic costs. The insight behind bilateral barter is that when ISPs take a holistic view of the traffic that they exchange, their interests are not completely opposed to each other, but all of them can gain simultaneously compared to flow-level anarchic routing. The motivation behind using agnostic costs is that they do not require ISPs to disclose sensitive information such as path latency or bandwidth. They also permit ISPs with diverse objectives to coordinate.

To evaluate Wiser, I used measured ISP topologies, realistic workloads, and two independent implementations. I considered several metrics of efficiency. For the scenarios where ISPs have comparable metrics, I considered two metrics – the length of Internet paths and a measure of the amount of bandwidth provisioning that ISPs require to deal with load variations, for example, due to failures. For the topologies and workloads that I studied, I found that the efficiency of Wiser is higher than *anarchy* which I model using currently common routing practices, and it is close to optimal routing which is a hypothetical scenario in which the Internet routing is globally optimized with complete information. Compared to optimal routing, the average path length is only 4% higher with Wiser and 13% higher with *anarchy*. While this average gain is useful, the key difference is in the tail of the path length distribution. The worst 1% of the paths are 6 times longer with *anarchy* but only 1.5 times longer with Wiser. For the bandwidth metric, Wiser reduces ISPs' provisioning requirements by 8% on average compared to *anarchy*. For the scenarios where ISPs have incomparable metrics, the efficiency of Wiser is close to being Pareto-optimal, and Wiser enables ISPs to cooperate such that each gains according to its own objectives.

For the alternatives that I considered, approaches that are potentially simpler than Wiser do not lead to similar efficiency. I found that a holistic barter for traffic exchanged between adjacent ISPs is key to high efficiency when win-win routing is desired. I also found that ordinal preferences, which disclose less information than the cardinal preferences of Wiser, lead to poor efficiency.

I implemented Wiser in SSFNet [113] and XORP [129] and found that it is easy to implement: starting from existing BGP implementations, it required less than 6% additional lines of code. The routing message processing overhead that Wiser imposes on routers is similar to BGP, and for workloads seen by routers today, its computation overhead is within 15 to 25% of BGP.

Finally, for the topologies, workloads and strategies that I studied, I found that the cost normalization and virtual payment ratio constraints in Wiser limit the gains that a cheating ISP can achieve.

8.1 Thesis and Contributions

The thesis demonstrated by my dissertation is that *a protocol based on bilateral barter between adjacent ISPs that act in their own interest can lead to efficient routing in the Internet and can be practically implemented.* By efficient, I mean that when ISPs use comparable metrics the routing quality is close to optimal routing which is globally optimized with complete information. By practical, I mean that the protocol preserves ISP autonomy and its complexity, measured in terms of implementation, routing message, and processing requirements, is comparable to that of today's protocols.

The key contributions of my work are:

A novel approach for Internet routing with competing interests Wiser is based on a novel approach that combines bilateral barter and agnostic costs. Bilateral barter takes a holistic view of traffic exchanged between two adjacent ISPs, which enables efficient and win-win routing. Agnostic costs, or cardinal preferences, enable ISPs with diverse objectives to coordinate and limit the amount of information that ISPs are required to disclose. My evaluation shows that the combination of the two produces routing that is almost as efficient as potentially more complicated approaches based on multilateral coordination or global currency. It also suggests that simplifying my approach further would reduce efficiency.

A practical and efficient Internet routing protocol To my knowledge, Wiser is the first protocol that is both practical and leads to efficient routing between ISPs. Wiser preserves ISP autonomy and has low overhead. It can be deployed in a framework that is similar to the current routing protocol. It retains today's simple monetary exchange practices in which payments between ISPs are coarsely tied to the amount of traffic exchanged and independent of the direction of the traffic. It also retains the current pair-wise contractual structure in which only neighboring ISPs have contracts with each other. Finally, Wiser is incrementally deployable in that two adjacent ISPs can use it to improve routing between them without waiting for deployment by other ISPs.

Understanding the impact of anarchy and autonomy in the Internet I combine empirical evaluation with analysis to understand the impact of anarchy and autonomy in the Internet. I show that while the efficiency of anarchy in the Internet today is acceptable on average, perhaps due to network engineering by ISPs, it is poor for a small fraction of paths. The unreliability and operational cost associated with the manual control required to fix this tail suggests that the price of anarchy is high in the Internet. I also show that, for the topologies, workloads, and ISPs' behaviors that I study, the efficiency of Wiser is uniformly high, suggesting that the price of autonomy is low.

To gain insight into the empirical results, I use simple analytic models to compute the efficiency of Wiser as a function of ISPs' internal costs of carrying traffic. The analysis predicts that Wiser is efficient when ISP costs are similar but inefficient otherwise. With dissimilar ISP costs, the efficiency is low because of the win-win requirement. That the efficiency of Wiser is high in practice suggests that the costs of ISPs that interconnect in multiple places tend to be similar.

8.2 Future Work

Three promising directions for future research arise from the work presented in this dissertation. I discuss these directions in this section.

8.2.1 Incremental deployment of Wiser

Many design decisions in Wiser have been guided by the need to keep the protocol practical and to simplify deployment. But since the proof is in the pudding, an immediate area of future work is to push for a deployment in the Internet. This is a four-step process which I outline below.

1. Enlist the support of two or more ISPs that interconnect in multiple places and who are willing to experiment with Wiser as a traffic optimization tool. In some cases, the same parent organization operates multiple ISPs; while it is by no means essential, enlisting such ISPs provides an easier testing ground with fewer competitive concerns.

2. Build an emulation tool that predicts the impact of using Wiser to control the routing between these ISPs. An instance of the tool runs for each ISP's network. Each instance outputs the paths that various traffic flows would use if the ISPs were running Wiser, based on real-time information on the state of their networks and on information received from the other ISPs' instances. The network state information needed as input to this tool depends on the optimization goal. For instance, if an ISP wants to minimize average delay, the tool needs the latencies of network links. Network

operators consider the output of the tool as suggestions and may choose to implement some or all of them by re-configuring their routers. Such an emulation tool can be built using existing routing platforms [130, 129] and network measurement tools [111].

This emulation phase before a live implementation is important because operators are apprehensive of any new technology that changes routing, lest it hurt rather than help performance. The emulation tool would help build confidence in Wiser in a controlled setting. It will also be useful towards revising the design of Wiser based on the experiences of these operators.

3. Connect the emulation tool to network routers such that it automatically configures the path selection of routers as if they were directly running Wiser. Researchers have demonstrated that such a logically centralized control of ISP networks is possible even for large tier-1 networks [25]. At this point, based on the experiences of these ISPs, more ISPs may be willing to run Wiser. This would create islands of ISPs that run Wiser; the islands get bigger as more ISPs join.

4. If there is enough market demand, router vendors will implement Wiser in their products, allowing for a router-level deployment.

I consider the exercise above to be a case study for understanding technology adoption in the Internet. Deploying new technologies in the Internet has proven to be a significant challenge [7, 94], and the forces that impact adoption are not always apparent. The experience gathered from the exercise above can be leveraged to develop simple, broadly applicable guidelines for improving the chances of a system being deployed.

8.2.2 Toolkit for managing competition in networked systems

One of the broader aims of my work is to use the concrete context of Internet routing to draw lessons for designing protocols for a wide class of competitive-yet-cooperative systems. An important step in this direction is to develop a toolkit of techniques that system designers can use to manage competition in networked systems, such that managing competition is as straightforward as ensuring reliable communication is today. Based on my experience with the design of Wiser, I make several observations on how to simplify protocol design in a competitive environment.

1. In many systems there is a conflict between efficiency and robustness to manipulation. Designs that are strictly robust to manipulation tend to be inefficient, and those that attempt to achieve maximal efficiency tend to be vulnerable to manipulation. In reality, parties in a system can be willing to cooperate beyond what might fall out of a strict game theoretic notion of rationality. For instance, ISPs tend to cooperate today using ad hoc mechanisms that are not exactly rational. Explicitly accounting for this cooperative behavior can increase efficiency and simplify system design because it allows for a lower level of protection against manipulation. Other researchers have made a similar observation in the context of peer-to-peer and wireless networks [51, 71].

2. Another approach for handling the conflict above is to increase the robustness to manipulation, without giving up efficiency, by reducing the degrees of freedom (or, in game theoretic parlance, strategy space) of individual parties. While not strictly foolproof, this may be sufficient in practice when it significantly reduces the gain a cheater can achieve.

3. Offline contracts between organizations can be leveraged to simplify online system operation. Because laws can be used to punish violations, most organizations will not

violate such contracts unless there is a huge win. Offline contracts can reduce complexity by i) reducing the number of available options [13]; ii) allowing behavioral verification at longer time scales, which is both more reliable and provides short-term flexibility to deal with special situations, as is done in Wiser; and iii) obviating the need for automated enforcement.

4. The normalization mechanism of Wiser enables trading between parties that want to trade as equals without disclosing their true valuation of the traded commodity. This ability is generally useful and can be used in other systems as well. For instance, it can be used to implement barter in a peer-to-peer file sharing network.

The first three observations above are germane not only to system designers but also to game theoreticians. Each suggests an enhancement to game theoretic models that will make them more amenable to the design and analysis of networked systems [70]. The suggested enhancements include relaxed notions of rationality and robustness to manipulation, and accounting for the impact of offline contracts on the online strategy space of parties.

8.2.3 Enhancements to Wiser

Finally, there are three important directions in which Wiser itself can be extended.

End-to-end quality-of-service (QoS) Wiser limits the amount of information that flows across ISP boundaries, but this also hinders cooperating ISPs from providing end-to-end QoS. An interesting avenue for future research is providing end-to-end QoS while limiting the information that an ISP discloses to others. My experiments with heterogeneous ISP objectives suggest that when metrics of end-to-end interest are used as one of possibly several factors that determine agnostic costs, the end-to-end paths are of good quality. This suggests that it might be possible to specify guidelines to derive agnostic costs that, if followed by ISPs, lead to end-to-end QoS. ISPs that follow these guidelines trade-off some autonomy in how they derive agnostic costs with providing interdomain QoS support to their customers. An important aspect of this effort entails understanding this trade-off.

Stability Stability of adaptive routing, especially in large-scale networks and using practical protocols, is a long-standing question of interest [44, 59, 125, 122, 6, 14, 106]. While overprovisioning in today's Internet suggests that Wiser will be stable under most perturbations, extending the protocol to be provably stable under all perturbations is an attractive area for future work. Recent work suggests that explicit feedback based control is a promising direction towards this goal [57, 56].

Quickly detecting cheating The robustness to cheating in Wiser is predicated on several factors. An important one is that, because ISPs value their reputation, they will not cheat if there is a chance that they will be caught. While I expect persistent cheating to be eventually detected by other ISPs (e.g., see reference [78]), automated cheating detection mechanisms will bolster deterrence. There is much ancillary information in the system that can be leveraged. For instance, a substantial disconnect between the paths that an ISP uses inside its network (which can be observed externally) and its announced costs indicates that the ISP is being dishonest about its cost. Similarly, an ISP's neighbors can cooperate and verify if the ISP announces similar costs to them at places where they both interconnect with that ISP. If the internal paths taken by traffic inside an ISP are known, as is often the case today, yet another possibility is to verify that the cost of destinations that use the same path is largely similar.

8.3 Summary

My work is a step towards Clark's vision of designing tools that recognize and leverage competing interests in the Internet [29, 30]. Wiser allows ISPs to find efficient routing paths, and thus offers better performance to their customers, while maintaining their autonomy. I showed how this can be accomplished within the current contractual framework between

ISPs and with only small changes to the current routing protocol. I hope that the lessons from my work will simplify the task of managing competition in other contexts as well.

BIBLIOGRAPHY

[1] Daniel J. Abadi, Don Carney, Uğur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Aurora: a new model and architecture for data stream management. *The VLDB Journal – The International Journal on Very Large Data Bases*, 12(2), August 2003.

[2] Eytan Adar and Bernardo A. Huberman. Free riding on Gnutella. *First Monday*, 5(10), October 2003.

[3] Mike Afergan and John Wroclawski. On the benefits and feasibility of incentive based routing infrastructure. In *Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems (PINS)*, September 2004.

[4] Aditya Akella, Jeff Pang, Bruce Maggs, Srinivasan Seshan, and Anees Shaikh. A comparison of overlay routing and multihoming route control. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2004.

[5] David Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient overlay networks. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, October 2001.

[6] Eric J. Anderson and Thomas Anderson. On the stability of adaptive routing in the presence of congestion control. In *Proceedings of the IEEE Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2003.

[7] Thomas Anderson, Larry Peterson, Scott Shenker, and Jonathan Turner. Overcoming the Internet impasse through virtualization. *IEEE Computer*, 38(4), April 2005.

[8] George Apostolopoulos, Roch Guerin, Sanjay Kamat, and Satish K. Tripathi. Quality of service based routing: A performance perspective. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, September 1998.

[9] Robert J. Aumann and Lloyd S. Shapley. Long-term competition – a game-theoretic analysis. Working Paper WP-676, Department of Economics, University of California, Los Angeles, September 1992.

[10] Daniel O. Awduche, Angela Chiu, Anwar Elwalid, Indra Widjaja, and XiPeng Xiao. Overview and principles of Internet traffic engineering. Request for Comments RFC-3272, Internet Engineering Task Force (IETF), May 2002.

[11] Barun Awerbuch, Yossi Azar, and Amir Epstein. The price of routing unsplittable flow. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, May 2005.

[12] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, June 2002.

[13] Magdalena Balazinska, Hari Balakrishnan, and Mike Stonebraker. Contract-based load management in federated distributed systems. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, March 2004.

[14] Anindya Basu, Alvin Liu, and Sharad Ramanathan. Routing using potentials: A dynamic traffic aware routing algorithm. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2003.

[15] Tony Bates, Ravi Chandra, and Enke Chen. BGP route reflection - an alternative to full mesh IBGP. Request for Comments RFC-2796, Internet Engineering Task Force (IETF), April 2000.

[16] Michel Berkelaar. lp_solve: linear programming code. ftp://ftp.ics.ele.tue.nl/pub/lp_solve/.

[17] Supratik Bhattacharyya, Christophe Diot, Jorjeta Jetcheva, and Nina Taft. PoP-level and access-link-level traffic dynamics in a Tier-1 PoP. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2001.

[18] Preeti Bhoj, Sharad Singhal, and Sailesh Chutani. SLA management in federated environments. Technical Report HPL-98-203, Hewlett-Packard Labs, December 1998.

[19] Olivier Bonaventure, Stefaan De Cnodder, Jeffrey Haas, Bruno Quoitin, and Russ White. Controlling the redistribution of BGP routes. Work in progress Internet draft: draft-ietfptomaine-bgp-redistribution-02, Internet Engineering Task Force (IETF), February 2003.

[20] Robert Braden, David Clark, and Scott Shenker. Integrated services in the Internet architecture: An overview. Request for Comments RFC-1633, Internet Engineering Task Force (IETF), June 1994.

[21] Steven J. Brams. *Negotiation Games: Applying game theory to bargaining and arbitration*. Routeledge, 1990.

[22] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the CONFIDANT protocol: Cooperation of nodes — fairness in dynamic ad-hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2002.

[23] Levente Buttyán and Jean-Pierre Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications*, 8(5), October 2003.

[24] Rajkumar Buyya, Heinz Stockinger, Jonathan Giddy, and David Abramson. Economic models for management of resources in peer-to-peer and grid computing. In *Proceedings of the SPIE International Symposium on The Convergence of Information Technologies and Communications (ITCom)*, August 2001.

[25] Matthew Caesar, Donald Caldwell, Nick Feamster, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. Design and implementation of a routing control platform. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, May 2005.

[26] Ravi Chandra, Paul Traina, and Tony Li. BGP communities attribute. Request for Comments RFC-1997, Internet Engineering Task Force (IETF), August 1996.

[27] Jianjun Chen, David J. DeWitt, Feng Tian, and Yuan Wang. NiagaraCQ: a scalable continuous query system for Internet databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, January 2000.

[28] Center for International Earth and Science Information Network. http://www.ciesin.columbia.edu.

[29] David Clark. The design philosophy of the DARPA Internet protocols. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 1988.

[30] David Clark, John Wroclawski, Karen Sollins, and Robert Braden. Tussle in cyberspace: Defining tomorrow's Internet. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2002.

[31] Bram Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, May 2003.

[32] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. How much can taxes help selfish routing? In *Proceedings of the ACM Conference on Electronic Commerce*, June 2003.

[33] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. Pricing network edges for heterogeneous selfish users. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, June 2003.

[34] Eric S. Crawley, Raj Nair, Bala Rajagopalan, and Hal Sandick. A framework for QoS-based routing in the Internet. Request for Comments RFC-2386, Internet Engineering Task Force (IETF), August 1998.

[35] Artur Czumaj and Berthold Vöcking. Tight bounds for worst-case equilibria. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, January 2002.

[36] Nick Feamster, Hari Balakrishnan, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. The case for separating routing from routers. In *Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, August 2004.

[37] Nick Feamster, Jay Borkenhagen, and Jennifer Rexford. Guidelines for interdomain traffic engineering. *ACM SIGCOMM Computer Communication Review (CCR)*, 33(5), October 2003.

[38] Joan Feigenbaum, Christos Papadimitriou, Rahul Sami, and Scott Shenker. A BGP-based mechanism for lowest-cost routing. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, July 2002.

[39] Joan Feigenbaum, Christos Papadimitriou, and Scott Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1), September 2001.

[40] Joan Feigenbaum and Scott Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, September 2002.

[41] Donald Ferguson, Christos Nikolaou, Jakka Sairamesh, and Yechiam Yemini. Economic models for allocating resources in computer systems. In Scott Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, January 1996.

[42] Bernard Fortz and Mikkel Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of the IEEE Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2000.

[43] Ian T. Foster and Carl Kesselman. Computational grids. In *Proceedings of the International Meeting on Vector and Parallel Processing (VECPAR)*, June 2000.

[44] Robert G. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Computers*, 25(1), January 1977.

[45] Lixin Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking (ToN)*, 9(6), December 2001.

[46] Vijay Gill. Private Communication, November 2003.

[47] Oded Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, August 2004.

[48] Timothy Griffin and Brian Premore. An experimental analysis of BGP convergence time. In *Proceedings of the International Conference on Network Protocols (ICNP)*, November 2001.

[49] David Hales and Simon Patarin. Computational sociology for systems "in the wild": The case of BitTorrent. In *Proceedings of the IEEE Distributed Systems Online*, July 2005.

[50] Mark Handley, Eddie Kohler, Atanu Ghosh, Orion Hodson, and Pavlin Radoslavov. Designing extensible IP router software. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, May 2005.

[51] Yang hua Chu and Hui Zhang. Considering altruism in peer-to-peer Internet streaming broadcast. In *Proceedings of the ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, June 2004.

[52] Internap Flow Control Xcelerator. http://www.internap.com/product/technology/fcx/.

[53] Philip Jacob and Bruce Davie. Technical challenges in the delivery of interprovider QoS. *IEEE Communications Magazine*, 43(6), June 2005.

[54] Ramesh Johari and John N. Tsitsiklis. Routing and peering in a competitive Internet. Technical Report P-2570, MIT LIDS, January 2003.

[55] Kirk Johnson, John Carr, Mark Day, and Frans Kaashoek. The measured performance of content distribution networks. In *Proceedings of the International Web Caching and Content Delivery Workshop*, May 2000.

[56] Srikanth Kandula, Dina Katabi, Bruce Davie, and Anna Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2005.

[57] Dina Katabi, Mark Handley, and Charles Rohrs. Internet congestion control for future high bandwidth-delay product environments. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2002.

[58] Alexander Keller and Heiko Ludwig. The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of System and Network Management*, 11(1), March 2003.

[59] Atul Khanna and John Zinky. The revised ARPANET routing metric. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, September 1989.

[60] Scott Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598), May 1983.

[61] Elias Koutsoupias and Christos H. Papadimitriou. Worst-case equilibria. In *Proceedings of the Symposium on Theoretical Aspects in Computer Science (STACS)*, March 1999.

[62] Basil Kruglov. Re: Cogent and level3 peering issues. NANOG mailing list archives: http://www.merit.edu/mail.archives/nanog/2002-12/msg00379.html, December 2002.

[63] Seth M. Kusiak. Re: Congestion peering C&W <-> @home. NANOG mailing list archives: http://www.merit.edu/mail.archives/nanog/2001-11/msg00282.html, November 2001.

[64] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. An experimental study of delayed Internet routing convergence. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2000.

[65] Kevin Lai, Michal Feldman, Ion Stoica, and John Chuang. Incentives for cooperation in peer-to-peer networks. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[66] Anukool Lakhina, John Byers, Mark Crovella, and Ibrahim Matta. On the geographic location of Internet resources. *IEEE Journal on Selected Areas in Communications (JSAC)*, 21(6), August 2003.

[67] Kirk Lougheed and Yakov Rekhter. A border gateway protocol (BGP). Request for Comments RFC-1105, Internet Engineering Task Force (IETF), June 1989.

[68] Sridhar Machiraju and Randy Katz. Reconciling cooperation with confidentiality in multi-provider distributed systems. Technical Report UCB-CSD-4-1345, Computer Science Division, University of California, Berkeley, August 2004.

[69] Sridhar Machiraju and Randy H. Katz. Verifying global invariants in multi-provider distributed systems. In *Proceedings of the Workshop on Hot Topics in Networks (HotNets)*, November 2004.

[70] Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Experiences applying game theory to system design. In *Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems (PINS)*, September 2004.

[71] Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Encouraging cooperation in multi-hop wireless networks. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, May 2005.

[72] Ratul Mahajan, Neil Spring, David Wetherall, and Thomas Anderson. Inferring link weights using end-to-end measurements. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.

[73] Ratul Mahajan, David Wetherall, and Thomas Anderson. Understanding BGP misconfiguration. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2002.

[74] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating router misbehavior in mobile ad-hoc networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, August 2000.

[75] Danny McPherson and Vijay Gill. BGP MED considerations. Work in progress Internet draft: draft-ietf-grow-bgp-med-considerations-04, Internet Engineering Task Force (IETF), June 2005.

[76] CA*net routing policy. http://www.canarie.ca/canet4/services/c4_routing_policy.pdf, March 2003.

[77] Alberto Medina, Nina Taft, Kavé Salamatian, Supratik Bhattacharyya, and Christophe Diot. Traffic matrix estimation: Existing techniques and new directions. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2002.

[78] Ron Miller. Legal battle ended for AT&T, MCI. InternetNews.com, February 2004. http://www.internetnews.com/xSP/article.php/3316751.

[79] Richard Mortier and Ian Pratt. Incentive based inter-domain routeing. In *Proceedings of the Internet Charging and QoS Technology Workshop*, September 2003.

[80] Jon Moy. OSPF version 2. Request for Comments RFC-2178, Internet Engineering Task Force (IETF), July 1997.

[81] Katta Murty. *Linear Programing*. John Wiley & Sons, 1983.

[82] Roger B. Myerson and Mark A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2), April 1983. Cited in Brams [21].

[83] NANOG mailing list. http://www.nanog.org/mailinglist.html.

[84] Samphel Norden and Jonathan Turner. Inter-domain QoS routing algorithms. Technical Report WUCS-02-06, Washington University, Department of Computer Science, January 2002.

[85] William B. Norton. Internet service providers and peering. Equinix whitepaper, version 2.5, May 2001. http://www.equinix.com/pdf/whitepapers/PeeringWP.2.pdf.

[86] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2001.

[87] Christos Papadimitriou. Algorithms, games, and the Internet. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, July 2001.

[88] Jian Qiu, Ruibing Hao, and Xing Li. An experimental study of the BGP rate-limiting timer. In *IEEE 4th International Network Conference (INC)*, July 2004.

[89] Lili Qiu, Yang Richard Yang, Yin Zhang, and Scott Shenker. On selfish routing in Internet-like environments. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2003.

[90] Bruno Quoitin, Cristel. Pelsser, Olivier Bonaventure, and Steve Uhlig. A performance evaluation of BGP-based traffic engineering. *International Journal of Network Management*, 15(3), February 2005.

[91] Bruno Quoitin, Sébastien Tandel, Steve Uhlig, and Olivier Bonaventure. Interdomain traffic engineering with redistribution communities. *Computer Communications Journal*, 27(4), March 2004.

[92] Barath Raghavan and Alex C. Snoeren. Priority forwarding in ad hoc networks with self-interested parties. In *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*, June 2003.

[93] Howard Raiffa. *The art and science of negotiation*. Harvard University Press, November 1982.

[94] Sylvia Ratnasamy, Scott Shenker, and Steven McCanne. Towards an evolvable Internet architecture. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2005.

[95] Yakov Rekhter and Tony Li. A border gateway protocol 4 (BGP-4). Request for Comments RFC-1771, Internet Engineering Task Force (IETF), March 1995.

[96] Eric C. Rosen, Arun Viswanathan, and Ross Callon. Multiprotocol label switching architecture. Request for Comments RFC-3031, Internet Engineering Task Force (IETF), January 2001.

[97] Tim Roughgarden. Designing networks for selfish users is hard. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, October 2001.

[98] Tim Roughgarden. Stackelberg scheduling strategies. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, July 2001.

[99] Tim Roughgarden. Selfish routing with atomic players. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, January 2005.

[100] Tim Roughgarden and Eva Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2), March 2002.

[101] RouteScience Path Control. http://www.routescience.com.

[102] AT&T route server. telnet://route-server.ip.att.net.

[103] RouteViews project. http://www.routeviews.org, January 2005.

[104] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking (MMCN)*, January 2002.

[105] Stefan Savage, Andy Collins, Eric Hoffman, John Snell, and Thomas Anderson. The end-to-end effects of Internet path selection. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 1999.

[106] Anees Shaikh, Jennifer Rexford, and Kang G. Shin. Load-sensitive routing of long-lived IP flows. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, September 1999.

[107] Scott Shenker, David Clark, Deborah Estrin, and Shai Herzog. Pricing in computer networks: Reshaping the research agenda. *ACM SIGCOMM Computer Communication Review (CCR)*, 26(2), April 1996.

[108] Jeffrey Shneidman, David C. Parkes, and Laurent Massoulie. Faithfulness in Internet algorithms. In *Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems (PINS)*, September 2004.

[109] Neil Spring, Ratul Mahajan, and Thomas Anderson. Quantifying the causes of path inflation. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2003.

[110] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking (ToN)*, 12(1), February 2004.

[111] Neil Spring, David Wetherall, and Thomas Anderson. Reverse engineering the Internet. In *Proceedings of the Workshop on Hot Topics in Networks (HotNets)*, November 2003.

[112] Vikram Srinivasan, Pavan Nuggehalli, Carla F. Chiasserini, and Ramesh R. Rao. Cooperation in wireless ad hoc networks. In *Proceedings of the IEEE Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2003.

[113] Scalable simulation framework. http://www.ssfnet.org/.

[114] Sam Stickland. Utilising upstream MED values. NANOG mailing list archives: http://www.merit.edu/mail.archives/nanog/2005-03/msg00400.html, March 2005.

[115] Michael Stonebraker, Paul M. Aoki, Witold Litwin, Avi Pfeffer, Adam Sah, Jeff Sidell, Carl Staelin, and Andrew Yu. Mariposa: A wide-area distributed database system. *The VLDB Journal – The International Journal on Very Large Data Bases*, 5(1), January 1996.

[116] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proceedings of the IEEE Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, June 2002.

[117] Hongsuda Tangmunarunkit, Ramesh Govindan, and Scott Shenker. Internet path inflation due to policy routing. In *Proceedings of the SPIE ITCOM Workshop on Scalability and Traffic Control in IP Networks*, August 2001.

[118] Hongsuda Tangmunarunkit, Ramesh Govindan, Scott Shenker, and Deborah Estrin. The impact of routing policy on Internet paths. In *Proceedings of the IEEE Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2001.

[119] Steve Uhlig and Olivier Bonaventure. Designing BGP-based outbound traffic engineering techniques for stub ASes. *ACM SIGCOMM Computer Communication Review (CCR)*, 34(5), October 2004.

[120] Steve Uhlig and Bruno Quoitin. Tweak-it: BGP-based interdomain traffic engineering for transit ASes. In *Proceedings of the Conference on Next Generation Internet Networks Traffic Engineering (NGI)*, April 2005.

[121] Curtis Villamizar, Ravi Chandra, and Ramesh Govindan. BGP route flap damping. Request for Comments RFC-2439, Internet Engineering Task Force (IETF), November 1998.

[122] Srinivas Vutukury and J.J. Garcia-Luna-Aceves. A simple approximation to minimum delay routing. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, September 1999.

[123] Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and W. Scott Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18(2), February 1992.

[124] Feng Wang and Lixin Gao. On inferring and characterizing Internet routing policies. In *Proceedings of the Internet Measurement Conference (IMC)*, October 2003.

[125] Zheng Wang and Jon Crowcroft. Analysis of shortest-path routing algorithms in a dynamic routing network environment. *ACM SIGCOMM Computer Communication Review (CCR)*, 22(2), April 1992.

[126] Zheng Wang and Jon Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications (JSAC)*, 14(7), September 1996.

[127] Jared Winick, Sugih Jamin, and Jennifer Rexford. Traffic engineering between neighboring domains. http://www.research.att.com/~jrex/papers/interAS.pdf, July 2002.

[128] Xipeng Xiao and Lionel M. Ni. Internet QoS: A big picture. *IEEE Network*, 13(2), March 1999.

[129] XORP: Open source IP router. http://www.xorp.org/.

[130] GNU Zebra – routing software. http://www.zebra.org/.

[131] Lixia Zhang, Stephen Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala. RSVP: a new resource ReSerVation protocol. *IEEE Network*, 7(5), September 1993.

[132] Yin Zhang, Matthew Roughan, Nick Duffield, and Albert Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, June 2003.

[133] Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of the IEEE Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2003.

[134] Sheng Zhong, Li (Erran) Li, Yanbin Grace Liu, and Yang Richard Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks – an integrated approach using game theoretical and cryptographic techniques. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, August 2005.

Appendix A

THE COST OF OPTIMAL ROUTING IN THE TWO-ISP MODEL

In this appendix, I outline the derivation of the expected cost of optimal routing under the two ISP model described in Section 6.2.1. Recall that the model consists of two ISPs, ISP-1 and ISP-W, interconnecting in N places. The internal topology of each ISP is a random mesh such that the average cost of transporting a packet between any two nodes is drawn from a uniform random distribution in the range [0..1] for ISP-1 and the range $[0..W]$ ($W \geq 1$) for ISP-W.

Optimal routing selects the interconnection that minimizes the total cost of a packet across the two ISPs. The cost of a packet through an interconnection can be modeled as the sum of two random numbers drawn from the appropriate ranges. The expected cost of optimal routing then is the expected value of the minimum sum.

I first derive an expression for the sum through an interconnection and then use it to compute the expected optimal cost. Let $F(s)$ be the cumulative distribution function (CDF) of the sum of the costs across an interconnection being s. Let y_1 and y_W be the two random numbers that represent the respective internal costs inside the two ISPs. Then, $F(s)$ is the probability that $y_1 + y_W \leq s$. To derive the expression for $F(s)$, consider three cases:

Case I: $0 \leq s \leq 1$ In this case $F(s)$ is simply $\int_0^s dy_1 \int_0^{s-y_1} \frac{dy_W}{W}$, as y_1 can vary between 0 and s and, having fixed y_1, y_W can vary between 0 and $s - y_1$. dy_W is divided by W because y_W is picked from the range $[0..W]$. Solving the integral yields $F(s) = \frac{s^2}{2W}$.

Case II: $1 \leq s \leq W$ The value of $F(s)$ for this case can be derived similarly except that y_1 can now vary between 0 and 1. Thus, $F(s) = \int_0^1 dy_1 \int_0^{s-y_1} \frac{dy_W}{W} = \frac{2s-1}{2W}$.

Case III: $W \leq s \leq W+1$ In this case, y_1 varies between 0 and 1, and y_W varies between 0 and the minimum of $s - y_1$ and W. $F(s) = \int_0^1 dy_1 \int_0^{min(s-y_1,W)} \frac{dy_W}{W}$. Assuming that $s = W + \delta$, where $0 \leq \delta \leq 1$, $F(s) = \int_0^1 dy_1 \int_0^{min(W+\delta-y_1,W)} \frac{dy_W}{W} = \int_0^\delta dy_1 \int_0^W \frac{dy_W}{W} + \int_\delta^1 dy_1 \int_0^{W+\delta-y_1} \frac{dy_W}{W}$. Solving the integral and substituting $\delta = s - W$ yields $F(s) = \frac{2s-1-(s-W)^2}{2W}$.

Combining the results from all three cases:

$$F(s) = \begin{cases} \frac{s^2}{2W} & 0 \leq s \leq 1 \\ \frac{2s-1}{2W} & 1 \leq s \leq W \\ \frac{2s-1-(s-W)^2}{2W} & W \leq s \leq W+1 \end{cases} \tag{A.1}$$

The expected cost of optimal routing can now be computed as:

$$C_{optimal}(N,W) = \int_0^{1+W} xNdF(x) \left(\int_x^{1+W} dF(y) \right)^{N-1} \tag{A.2}$$

The parenthetical term above computes the probability of the sum through $N-1$ interconnections being greater than x and $dF(x)$ is the probability that the sum through one of them is x. The product of the two and N is the probability that the minimum sum is x because any of the N interconnections can be optimal. Multiplication by x and integration yields the expected cost of optimal routing.

Because $dF(y)$ is not a continuous function, the computation requires integration by parts.

$$\begin{aligned} C_{optimal}(N,W) &= \int_0^1 xNdF(x) \left(\int_x^1 dF(y) + 1 - \frac{1}{2W} \right)^{N-1} \\ &+ \int_1^W xNdF(x) \left(\int_x^W dF(y) + \frac{1}{2W} \right)^{N-1} \\ &+ \int_W^{W+1} xNdF(x) \left(\int_x^{W+1} dF(y) \right)^{N-1} \end{aligned} \tag{A.3}$$

In the first term on the right hand side, $\int_x^1 dF(y)$ is the probability that the sum is between x and 1, and $1 - \frac{1}{2W}$ is the probability that the sum is greater than 1. Their combination yields the sum being greater than x, while integrating only over the range where $dF(y)$ is continuous. The other two parenthetical terms are similarly derived.

Substituting the appropriate expressions for $F(y)$ yields the value of the expected cost of optimal routing:

$$\begin{aligned} C_{optimal}(N,W) &= \frac{N}{3W}\, {}_2F1(\frac{3}{2}, 1-N, \frac{5}{2}, \frac{1}{2W}) \\ &+ \frac{N}{2(2W)^N}\left(\frac{(2W+1)((2W-1)^N-1)}{N} - \frac{(2W-1)^{N+1}-1}{N+1}\right) \\ &+ \frac{2WN+W+1}{(2N+1)(2W)^N} \end{aligned} \tag{A.4}$$

where ${}_2F1$ is the Gauss hyper-geometric function: ${}_2F1(a,b,c,z) = \sum_{k=0}^{\infty} \frac{\binom{a}{k}\binom{b}{k}z^k}{\binom{c}{k}k!}$.

Individual ISP contributions to the total cost can be computed in a similar manner. First, observe that the expected contribution from ISP-1 is $s/2$ when $s \leq 1$, is $1/2$ when $1 \leq s \leq W$, and is $(s-W+1)/2$ when $W \leq s \leq W+1$. Then, use these values in Equation A.3 to derive $C^1_{optimal}(N,W)$, the contribution of ISP-1.

$$\begin{aligned} C^1_{optimal}(N,W) &= N \times \int_0^1 \frac{x}{2} dF(x) \left(\int_x^1 dF(y) + 1 - \frac{1}{2W}\right)^{N-1} \\ &+ N \times \int_1^W \frac{1}{2} dF(x) \left(\int_x^W dF(y) + \frac{1}{2W}\right)^{N-1} \\ &+ N \times \int_W^{W+1} \frac{x-W+1}{2} dF(x) \left(\int_x^{W+1} dF(y)\right)^{N-1} \end{aligned} \tag{A.5}$$

Solving the above yields:

$$\begin{aligned} C^1_{optimal}(N,W) &= \frac{N}{6W}\ {}_2F1\left(\frac{3}{2}, 1-N, \frac{5}{2}, \frac{1}{2W}\right) \\ &+ \frac{(2W-1)^N - 1}{2(2W)^N} \\ &+ \frac{N+1}{(2N+1)(2W)^N} \end{aligned} \tag{A.6}$$

The expected cost inside ISP-W is simply:

$$C^W_{optimal}(N,W) = C_{optimal}(N,W) - C^1_{optimal}(N,W) \tag{A.7}$$

VITA

Ratul Mahajan received his Bachelor of Technology degree in Computer Science and Engineering from the Indian Institute of Technology, Delhi, India in 1999, and his Master of Science degree in Computer Science and Engineering from the University of Washington in 2001. His research interests lie in the area of networked computer systems, especially the architecture and design of large-scale systems.