# Applying the Repeated Game Framework to Multiparty Networked Applications

by

Michael Moïse Afergan
AB, Harvard College (2000)
SM, Harvard University (2000)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[September 2005]
August 2005



Author .....
Department of Electrical Engineering and Computer Science
August 31, 2005

Certified by .....
Dr. David Clark
Senior Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

# Applying the Repeated Game Framework to Multiparty Networked Applications

by

Michael Moïse Afergan
AB, Harvard College (2000)
SM, Harvard University (2000)

Submitted to the Department of Electrical Engineering and Computer Science
on August 31, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

## Abstract

This thesis presents repeated game analysis as an important and practical tool for networked application and protocol designers. Incentives are a potential concern for a large number of networked applications. Well-studied examples include routing and peer-to-peer networks. To the extent that incentives significantly impact the outcome of a system, system designers require tools and frameworks to better understand how their design decisions impact these incentive concerns.

Repetition is a prevalent and critical aspect of many networking applications and protocols. Most networked protocols and architectures seek to optimize performance over a longer timescale and many have explicit support for repetition. Similarly, most players in networked applications are interested in longer horizons, whether they be firms building a business or typical individuals trying to use a system. Fortunately, the study of repeated interaction between multiple self-interested parties, repeated games, is a well-understood and developed area of economic and game theoretic research. A key conclusion from that literature is that the outcome of the repeated game can differ qualitatively from that of the one-shot game. Nonetheless, the tools of repeated games have rarely if ever been brought to bear on networking problems.

Our work presents the descriptive and prescriptive power of repeated game analysis by making specific contributions to several relevant networking problems. The applications considered are inherently repeated in practice, yet our research is the first to consider the repeated model for each particular problem. In the case of inter-domain routing, we first show that user-directed routing (e.g., overlays) transforms routing into a meaningfully repeated game. This motivates us to consider protocols that integrate incentives into routing systems. In designing such a routing protocol, we again use repeated games to identify important properties including the protocol period and the format of certain protocol fields. Leveraging this insight, we show how it is possible to address the problem of the repeated dynamic and arrive at a more desirable outcome. In the case of multicast overlay networks, we show how repeated

games can be used to explain the paradox of cooperative user behavior. In contrast to prior models, our repeated model explains the scaling properties of these networks in an endogenous fashion. This enables meaningful examination of the impact architecture and protocol design decisions have on the system outcome. We therefore use this model, with simulation, to descry system parameters and properties important in building robust networks.

These examples demonstrate the important and practical insights that repeated game analysis can yield. Further, we argue that the results obtained in the particular problems stem from properties fundamental to networked applications – and their natural relationship with properties of repeated games. This strongly suggests that the tools and techniques of this research can be applied more generally. Indeed, we hope that these results represent the beginning of an increased use of repeated games for the study and design of networked applications.

Thesis Supervisor: Dr. David Clark
Title: Senior Research Scientist

# Dedication

This dissertation is dedicated to my father, Barry Afergan.

In my life, I have been blessed with many wonderful and caring teachers and supporters. Of course, my family has always been my biggest and best supporter and I am forever indebted to both of my parents, my brother, and my grandfather. However, no single person is more responsible for my realization of this PhD than my father.

Most likely, none of this would have been possible had my father not facilitated the initial spark – getting me a computer (now literally a museum piece) at an early age. Over time he's tolerated my monopolizing of the household computers and phone line, helped me land and negotiate my first consulting engagements, and probably did much more that I readily took for granted.

However, most important are the tools and lessons that my father imparted to me that have made this research effort possible. Throughout my life, he has helped me to discover what *my* goals are and then has helped me to achieve them. He has taught me what it means to put my head down and be dedicated, yet at the same time, he has always been quick to make sure I pick my head up and keep a healthy dose of perspective. My father has given me confidence when I most needed it, yet has never been shy to remind me when I was in danger of having too much confidence. He has always provided me an amazing degree of independence to explore and tackle problems on my own, yet has always amazed me with his ability to understand both me and the problem at hand when I come to him for advice.

Thanks, todah, et merci.

# Acknowledgments

First and foremost, I must thank the two individuals who had advised me the most in this research:

- Dave Clark: Thanks to Dave's invaluable support, I was able to explore my research interests freely, ultimately stumbling upon a topic that I absolutely loved. Dave has been fantastic in helping me find the keystones to enable my more interesting work and conclusions–and throughout the process, his advice continually improved both my research and argument structure.

- Rahul Sami: A talk Rahul gave my very first week of grad school played a major role in my decision to pursue game theory. Since then, he been a fantastic advisor and supporter, excellent in helping me successfully navigate through a variety of challenges – from tackling high-level problems and to scrubbing the details of proofs.

I am also indebted to the other individuals who have advised me along the way:

- John Wroclawski: John has served as a fantastic research advisor from the nascent days of my undergraduate thesis to the final days of my PhD. John has a fantastic (and sometimes infuriating) ability to find the weaknesses in any idea – no matter how many days, weeks, or even months I spent preparing to present to him. At the same time, with simple but fundamental suggestions, he is fantastic at helping grow those ideas that do have merit (on the few rare occasions that I am lucky to have some). To the extent that anyone finds my research interesting, I owe a majority of that to John and to what he has taught me over the years.

- Dina Katabi: Rivaled only by John, Dina is amazing in her ability to quickly grasp a problem and the weaknesses of my research. In addition to emphasizing the portions of my research that needed help, her advice on how to address these weaknesses and fresh perspective on the problems dramatically improved my research.

- Drew Fudenberg: Drew's class, which I took my first semester at MIT, cemented my desire to pursue research in game theory and opened my eyes to various areas of research still open in the field. At times I felt certain Drew would tire as this computer science student stumbled to grasp certain game theory concepts or showed up with yet another somewhat incomplete proof. However, he proved to be a fantastic advisor and teacher, and I left each and every meeting with him far more excited (and far far smarter) than when I arrived.

In addition to my formal advisors, I have been fortunate to have a great set of friends and colleagues at MIT who have helped me in numerous ways:

- Rob Beverly: No one at MIT impacted my time there more than Rob. Outside of the office, he has become a great friend, advisor, workout partner, and finder of fine BBQ. Inside the office we called home, Rob's extensive knowledge and background in networking; his ability to help me see through my early-stage, often incoherent, and usually incorrect musings; his creativity and fresh perspective; and his incredible patience shaped and aided my research tremendously. Perhaps most important, the Red Sox have never lost a game that I have watched with Rob.

- Steve Bauer: Steve is my longest friend at MIT. He's also one of my best – and most dependable – critics, always ready and eager to find a new hole in my ideas. His criticisms never fail to dramatically improve my research. And of course his deep thoughts on deep-sea submarine rescues, electronic toys, and many other not-so-serious topics have made my time at MIT much more enjoyable.

- Becky Shepardson: The number of things that Becky has aided me with over the years could fill the remaining 150 pages of this thesis itself. She keeps our group together and is always quick to provide much needed assistance – and at times comic relief.

- Karen Sollins: Karen is one of the nicest people I have had the pleasure of working with at MIT. She has always been happy to give advice or feedback on a variety of topics – from which grad school I should attend to how I should structure my PhD dissertation defense.

I would be remiss if I did not also thank George Lee, Arthur Berger, and Peyman Faratin for their help and friendship. Arthur had several significant contributions to the research that became Chapter 4. Peyman significantly aided and encouraged my initial explorations into game theory and served as a trusty swimming partner.

Thanks also to Bobby Kleinberg was always willing and ready to give great advice. Nate Kushman, Hariharan Rahul, Srikanth Kandula, and Sachin Katti also made my time at MIT more enjoyable – and improved my research with their quick, sharp, and dead-on research advice. Charlie O'Donnell and Ian Bratt are great friends as well, making my time on the 8th floor far more fun.

I have too many friends and colleagues at Akamai to thank individually. However, I am forever indebted to Danny Lewin's lessons on creativity and tenacity – on which I drew several times during the PhD process.

Outside of MIT, I am fortunate to have far too many other friends to thank in these acknowledgments, which are already far too long. All of these friends made the past two years a fantastically fun time and also kept me sane. Jeff Weinshenker gets special credit for his professional-quality pinch-hitting editing help.

It goes without saying that I must also thank the Red Sox for sweeping the St Louis Cardinals in the World Series. Given that I was completely useless during the Red Sox' entire playoff run, my research may have fatally stalled if I had to deal with another week of anguish and glee.

Last but, of course, not least, I must thank my immediate family – my mother (my first and best teacher) and father (my ultimate mentor and supporter), my brother Dan (my lifelong friend and wrestling partner) and my grandfather (my endless source of encouragement, humor, and Red Sox enthusiasm). I am far too poor a writer to try to capture the impact of their love and support over the past 27 years – so I'll simply say that I love you all very much.

# Contents

# List of Figures

# Chapter 1

# Introduction

The Internet and networked systems in general continue to grow in several dimensions – of particular interest are the number of applications on these networks and their importance in daily life. Individual users ascribe extreme significance to their email, web, and cell phone connectivity. Businesses not only depend on these services, but moreover some firms – such as ISPs – exist solely to facilitate these services. In these cases, the users' incentives to maximize their experience and/or firms' desires to maximize their profits can be significant. This in turn can impact how they use the relevant applications and protocols, how efficiently the application or set of applications operate, and even which applications continue to exist and which fail and disappear.

A byproduct of this is that for a large class of applications, incentives must be an additional first order concern in the design process. It is a tempting but ultimately naive approach to simply design an elegant or seemingly efficient system and then later worry about incentives. Users will act upon their incentives in the context of the particular system and its protocols. Therefore, decisions made by the application or system designer which *a priori* seem benign may therefore become vitally important. Worse, systems believed to perform well may significantly degrade in practice – and there is not necessarily a simple solution to the incentive problem that can be readily added to the system after the fact. Thus, while self-interest is not necessarily bad, it is vitally important to find tools that are effective in helping to identify and address these tensions via system design.

Game theory is an area which holds promise for producing such valuable tools. A "game" is a strategic interaction between multiple self-interested parties. Game theory provides a rich set of tools for understanding how such players may desire to act in practice and how the rules and structure of the environment can impact

these behaviors. Consequently, several researchers have recently begun to apply game theoretic tools to networking problems in an attempt to build more efficient and robust systems.

However, the application of game theory to networking applications is not simple. Like many other modeling tools, game theory therefore requires careful selection of the appropriate concepts – balancing the goals of creating a model that is simultaneously valid yet tractable. Further, Internet-style networking is a particularly difficult problem domain. Most applications seek to optimize multiple goals, some of which are even in tension with each other. Worse, in some applications the players may not agree on the goals of the system and in others, the goals may be constantly evolving and changing. Within these applications, we often find complex and intricate protocols, with multiple parameters often used in an imprecise fashion. Therefore, finding tools that are truly effective for networked applications can be challenging.

In this thesis, we examine one such tool that holds promise for providing significant aid for system design: repeated game analysis. Informally, repeated game analysis considers an interaction not as a single event but rather a sequence of similar events occurring over time; with the actions in one period potentially impacting the state of the world and the actions of players in future periods. We believe that **repeated game analysis is an important and practical tool for the design of networked applications** and support this claim by analyzing three specific important networking applications. Our analysis is motivated by three key reasons:

1. **Repetition is an inherent aspect of almost all networked problems.** Routing and congestion control are examples of processes which are constantly repeated in similar circumstances. Individual users repeatedly interact with the same networks, often to accomplish the same or a similar set of tasks. Further, in peer-to-peer environments, users repeatedly interact with the same, or behaviorally similar, users.

2. **Repeated games are a well-understood area of game theory.** The literature here provides appropriate tools and concepts for the analysis of repeated games. These concepts have been shown to be robust to a wide array of practical assumptions and these dynamics have been observed and documented in practice.

3. **Repetition can significantly alter the outcome of a game.** In particular, the outcome of repeated games can differ in kind, not merely in degree, from

the outcome of the particular stage-game. This is a critical, fundamental, and robust result of the repeated game theory literature. It is also, to some degree, intuitive – when an interaction is to be repeated, it is natural to believe that the strategies and outcomes can change significantly. This is the crucial reason why repeated games must be considered in domains, such as networking, where repetition is prevalent.

This thesis presents the first use of repeated games as a primary tool for networked application design. Our work contrasts significantly with most prior work and literature in the area of game theory applied to networking, which has focused on the one-shot game. By capturing the repeated dynamic, our models have a higher degree of fidelity, permitting us to appreciate relationships that do not exist in the one-shot game. Consequently, they uniquely permit us to descry important system parameters that are of practical importance and consequence.

## 1.1 A Fundamental Relationship Between Networked Applications and Repeated Games

The prevalence of repetition, as discussed above, suggests that repeated games are an appropriate model for networked applications. More importantly, in the applications considered in this thesis, our use of repeated-game models yields results with practical significance. This, we argue, stems from a fascinating and useful alignment between properties fundamental to networking applications and those fundamental to repeated games. These relationships make repeated games both important and practical for networked applications. Moreover, they suggest that the intuition and techniques of this thesis generalize to a large class of networking applications.

Networked applications are varied, but there are several properties which are fundamental to many networked applications. In the context of this thesis, there are four key relevant properties common to most networked applications. We first overview these properties and then explain their significance to repeated games.

1. **Networked Applications Involve Multiple Interacting Self-Interested Parties.** Networks, by definition, facilitate communication between multiple, often independent, parties. Often these interactions can also share infrastructure with other unknown parties. Further, as the Internet and other networks

become more commercialized, various firms have entered the milieu to provide additional services. These firms are, by definition, independent and self-interested.

2. **Interactions of Networked Applications are Repeated.** Networks are inherently built for operations and communications that are repeated. In some cases, such as routing, the repeated interaction is almost identical and between the same parties. In other cases, such as web-browsing or peer-to-peer networks, the parties in an interaction and the content exchanged can vary. However, the types of interactions, and in many cases the players involved can remain stable over a long period of time.

3. **Networked Applications Face Constraints.** These constraints stem from a number of sources. Some are related to the fundamental goal of the system, for example, a routing system will probably require relatively stable paths. Other potential constraints that can be driven by the application include privacy concerns, cost concerns, and/or a desire to eschew heavyweight mechanisms in the interest of flexibility and scalability. Still other constraints can come from the network. On the Internet, and in most networked systems, these typically include an inability for nodes to monitor each other, a lack of identities, or a lack of a mechanism for micro-payments.

4. **Actions within Networked Applications are Highly Parameterized.** Depending on the application, these parameters can govern a number of important factors, such as the period between rounds of the protocol, how a tree formation protocol should run, or a measure of the state of the network. Further, there are meta-parameters, such as how many bits we allocate to certain parameters in protocol headers. While the presence of many parameters may be seen both as desirable (for flexibility) or undesirable (for complexity), many of these parameters are unavoidable. Indeed, even a parameter that is set dynamically still impacts the system. In other cases, the value can be left unspecified, but that too is a setting of the parameter (often to be infinitely small or infinitely large).

Properties 1 and 2 suggest that repeated games are an essential consideration for networking applications. Property 1 is the motivation for much recent work applying game theoretic tools to networking applications. As discussed, repetition qualitatively impacts the actions of the players and the system outcome. Consequently, property

2 implies that any game theoretic analysis of a networked application *must* consider the impact of the repeated case.

In contrast to properties 1 and 2, property 3 suggests why repeated games are of practical importance and utility. These constraints significantly and meaningfully limit the space of architectures and thus mechanisms that can be employed. From a game theoretic perspective, this means even if a solution to the one-shot game exists, it may be impossible to realize the corresponding mechanism. For example, in Chapter 4, we explain why the strong one-shot game results of Feigenbaum *et al* [32] in the area of routing do not hold in the repeated game. In other cases, consideration of the repeated game may provide a simpler approach and/or may be the only explanation for behavior currently observed in practice. Instead of working hard to modify the problem statement, straining the constraints of what is practical, or worse – assuming away the constraints – a far more practical and useful approach may be to consider the repeated game as a simpler model, especially as it captures the players' natural concern for the future. For example, in Chapter 5, we show that modeling multicast application overlays as a repeated game presents a potential alternative to architectures that rely on heavy-weight mechanisms such as payment or identity systems.

The parameterized action space (Property 4) is significant in two different ways. It should not be surprising that in many cases, selection of an appropriate parameter *value* is important. With many parameters, there are thus many values that must be selected. For this, there are a number of tools, including game theory–as well as control theory and machine learning. In many cases, it is desirable for the system to dynamically select the parameter values.

However, there is also a far more fundamental and elegant relationship that is unique to parameterized action spaces and repeated games. In a networked protocol, values exchanged are inherently discrete. However, in the repeated game, the granularity of the action space has a qualitative impact on the equilibrium outcome[1]. For a networked protocol, therefore, the parameter *granularity* can significantly impact the system outcome and hence takes on newfound importance. Repeated game analysis can therefore help to understand which parameters are significant, how they impact the outcome, and in some cases in what direction the outcome moves as a parameter is changed. For example, in the routing example of Chapter 4, we show that the number of bits allocated to certain protocol fields can significant impact the system

[1]This is unlike the one-shot game where for the most part the granularity is rounding error. We discuss this in more detail in Section 2.5

outcome.

Taken together, while properties 1 and 2 motivate the research of this thesis, it is properties 3 and 4 which facilitate the insight and practical conclusions that are obtained. These four properties are not unique to the problems considered in this thesis, but rather general to a large class of networked applications. They thus suggest that repeated games may be a practical tool in many other instances. Indeed, we believe that the concepts and methodology presented in this thesis can be applied to a variety of networked applications, including but not limited to ad-hoc networks, several general wireless networks, and a large number of peer-to-peer problems.

## 1.2 Research and Thesis Overview

While the techniques are general, this thesis explores the application of repeated games to specific networking problems. Each example is offered to demonstrate the benefits of repeated games in a different context. For each problem, we seek to understand the dynamics at play, how the system parameters and design decisions impact the system outcome, and how such decisions and parameters can be used to build more robust, stable, and/or efficient systems. In doing so, we discover key insights into the individual problems, uniquely facilitated by the tool of repeated games. This approach contrasts with prior work not only in the fact that we look at the repeated game but also because we do not necessarily attempt to achieve a particular social choice function (e.g., a strategyproof mechanism) since such a result may be impractical and/or undesirable. (We discuss this further in Section 6.2.)

Before beginning with the individual problems, Chapter 2 presents a set of background material. It reviews the relevant tools from game theory and repeated game theory. It also discusses some related work that spans the multiple problems. (Each problem chapter presents the related work specific to that problem.) After this background, the thesis examines three important problems involving networked applications.

Chapter 3 uses *repeated games as a higher-level modeling tool* to understand more fully recent changes in the dynamics of inter-network routing and to motivate a design approach. User-directed routing, currently in the form of overlays, peer-to-peer networks, and potentially in the form of source-based routing, transform the stability and thus the nature and action-space of inter-network relationships. Recognizing this, we examine models of routing and see that enhancing the models with the notion of the relevant repeated game delineates a problem that was previously not well-understood.

We argue that this analysis warrants consideration of a design architecture where pricing and routing are more logically coupled, and present a set of principles for how to do so.

Chapter 4 uses *repeated games to produce a formal model which permits insightful analysis* to answer a specific protocol-design challenge. Motivated by the notion of coupling pricing and routing, we ask the question, "How should one design a protocol to convey pricing information for routes?" We present previously unrecognized problems that may result in the context of routing. To understand when and how these problems may arise, and to enable protocol designers to understand how their design decisions impact this behavior, we model and analyze the problem as a repeated game. This allows us to descry several system parameters that are very important, but which *a priori* seem insignificant. Because these results relate to specific protocol parameters, they are thus directly applicable to routing protocol design.

Chapter 5 uses repeated games as a *natural motivation* and *endogenous modeling tool* to explain the dynamics of application overlay multicast networks and then *simulates* the game to understand how to build more robust networks. The efficiency of application overlay multicast networks comes from the tree-formation protocol used. However, in practice, user-nodes have both the means and motive to alter the structure of the tree to improve their experience. This can be done by moving up the the tree, which may improve the quality of the incoming data, and/or by not supporting children, which may consume CPU and bandwidth. We formalize this problem, and show that it is hard to resolve without using very heavyweight mechanisms such as payment or verification schemes. These mechanisms don't exist today, and even if they did, their cost of implementation motivates the need to understand exactly where and when such machinery is needed and/or beneficial. However, even without such mechanisms, users are inherently interested in the continued existence of the network. This inherent notion is naturally captured in the the repeated model. The model therefore provides a simple (and practical) way to model cooperative behavior and contrasts with prior modeling as the incentive to cooperate is endogenous to the model. This in turn allows us to analyze how the software and protocol used can affect the efficiency of the network.

After presenting these examples, we step back in Chapter 6 to discuss the common themes and lessons from the examples and discuss repeated games as a general tool for system developers.

## 1.3 Key Contributions

As discussed, this thesis makes contributions at two levels. In three particular relevant networking problems, we obtain important new results. These results, coupled with our higher-level analysis suggest that repeated games can be a generally useful tool for networked application design. We summarize these contributions:

- Repeated Games as a Tool for Networked Applications
  - ◇ Demonstration of the applicability of repeated games to networked applications via three distinct examples. Further, each example demonstrates a different facet of repeated games as a tool.
  - ◇ An argument that the importance and practicality of repeated games stems from a fundamental relationship between repeated games and networked applications. This is significant as it suggests a broad applicability of the tools and techniques.
- Benefits and Feasibility of Incentive Based Routing
  - ◇ Demonstration that user-directed routing has transformed routing to the point that the repeated game model is the most appropriate.
  - ◇ Observation that when viewed via the lens of a repeated game model, inter-domain routing may lack the stability that is desirable from a system-design perspective.
  - ◇ Motivation for the design decision to couple routing and pricing mechanisms.
- The Design of Incentive-Based Routing Systems
  - ◇ Specific protocol parameters including the protocol period, the granularity of the format, and the width of the price field can have a significant on the outcome of the system.
  - ◇ These parameters have specific relationships which can be leveraged to control the system, if desired.
- Building Robust Application Overlay Networks

- ◇ A proof that underlying incentive problems coupled with practical problems such NAT-based Sybil attacks[2], prevents any algorithm from guaranteeing non-trivial efficiency.
- ◇ A model of cooperation, based on the repeated dynamics and users' concern for the future. This model is novel in that it makes cooperation endogenous to the model. As such, it provides a simpler explaination for observed phenomena and facilitates the comparison of alternative architectures.
- ◇ Practial results on how to build systems that scale more efficiently. We find that the inherent structure of NICE trees provides robustness in selfish environments. Further, we discover the importance of the cluster size parameter, which, under reasonable assumptions, can be used to improve the robustness of the system.

## 1.4 Bibliographic Notes

Parts of Chapter 3 appeared as *On the Benefits and Feasibility of Incentive Based Routing*, Mike Afergan and John Wroclawski, in *The Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, ACM Press, 2004 [1]. At the time of the submission of this thesis, parts of Chapters 4 and 5 are both under submission for publication. Chapter 5 is joint work with Rahul Sami.

[2]A Network Address Translator (NAT) is a networking device that enables many users to share on IP address. A Sybil attack is one in which a single user pretends to have many identities in a given system. NATs therefore are a particularly simple way of masking Sybil attacks. We discuss this in more detail in Section 5.3.3.

# Chapter 2

# Background Material

This chapter is designed to provide an overview of the basic game theoretic tools required by this thesis and of literature related to this thesis. This chapter is not designed to be a complete introduction or summary of game theory. Readers desiring more background on game theory can consult one of several texts that include a presentation of repeated games, such as [77] or [40]. (The latter text is more advanced.) A more concise presentation can be found in [66]. After this overview of game theory and repeated games, we survey some works from the game theory literature and some works from the Computer Science literature that are relevant to the overall thesis. Each individual chapter also contains a discussion of more work specifically related to that problem.

## 2.1 Game Theory Basics

A game is a strategic interaction between multiple independent players. Formally, a game is a set of definitions explaining how the player can act, and how she desires to act. To explain some of the most important concepts, consider a sample game: an auction. Here we assume that we have $N$ bidders, one seller, and exactly one good to sell. Each player submits a sealed bid in writing; and the bidder with the highest bid pays that price and receives the good. (This game is referred to as a first priced auction.)

With this auction as context, we now present some important terms and concepts:

- *Action Space*: The set of permissible moves for a player. In this auction, the action is the bid, $b_i$, and the action space is $\mathbb{R}^+$, the positive real numbers. We

typically use subscript $-i$ as the notation to represent the other players. In this context, $b_{-i}$ represents the vector of actions for all players other than $i$.

- *Type Space*: A type is information (relevant to the game) that is private and particular to a user. A type space is the set of possible types. In this game, the type is the bidder's valuation of the good, $v_i$, and the type space is $\mathbb{R}^+$.

- *Outcome*: The outcome is a fully specified result of the game. In this auction, an outcome would be a decision on who receives the good, and how much she pays.

- *Utility Function*: A utility function, $u(.)$, maps outcomes to valuations for each player. In this sample auction game, the utility is the difference between the player's bid and her valuation ($v_i$) if she has the highest bid; and zero otherwise. More formally, we can write:

$$u_i(b_i, b_{-i}) = \begin{cases} v_i - b_i, & i \text{ wins the auction} \\ 0, & \text{otherwise} \end{cases}$$

- *Information Model*: This describes what the users know about each other and the game itself. In this game, this could include a model of the valuations other players or a model of how well the players really understand the good being sold.

A *strategy* specifies how a player will play given a particular type and state of the game (amongst other potential inputs). More formally, a strategy function in this game could map valuations to bids (i.e., s: $\mathbb{R}^+ \rightarrow \mathbb{R}^+$).[1] For example, a strategy which says "Bid half of what I value the good to be." could be written as:

$$s(v_i) = \frac{v_i}{2}$$

Here $s(.)$ is the strategy function which takes $v_i$ (the valuation) as its only input. The output is the bid, that is $b_i = s(v_i)$.

A *Nash equilibrium* of a game is a set of strategies such that no player wishes to change her strategy, given the strategy of the other players. More formally, $s$ is a Nash Equilibrium (NE) iff:

[1]Depending on the model, the strategy function may also depend on such factors as beliefs about the valuations (and/or bids) of the other players.

$$u_i(s_i, s_{-i}) \geq u(\hat{s}_i, s_{-i}) \; \forall \hat{s} \in \Omega$$

where $\Omega$ is the set of possible strategies and $u_i(s_i, s_{-i})$ is the utility realized by player $i$ when she plays $s_i$ and the other players play $s_{-i}$.

## 2.2 Repeated Game Overview

Unlike the simple auction example, most interactions in networking – and in other aspects of life – are repeated. As discussed in the Introduction, repeated games are therefore an important tool to accurately capture the true dynamics of a problem. Fortunately, repeated games are a well studied sub-field of game theory. While repeated games have a full and rich literature, we focus here on the basic concepts and tools that will prove important in this thesis. (We review some of the literature in Section 2.6.1.)

A repeated game is the repeated play of a particular one-shot game by the same players[2]. In the context of a repeated game, the particular one-shot game is called a *stage-game.*

Consider, for example, the canonical example of the Prisoners' Dilemma. In this game we have two players. The action space of this game is simple – each player can either cooperate (C) or defect (D), and the players move simultaneously. Based on their actions, each player receives a payoff as given by the matrix in Table 2.1. In the matrix, each cell represents the payoff of a particular pair of actions. For example, if both players play C then both get a payoff of 1, or $u_1(C, C) = u_2(C, C) = 1$. If however, player 1 plays D but player 2 plays C, then the payoff is $u_1(D, C) = 2$.

Table 2.1: Game Payoffs for the Prisoners' Dilemma

| P1 \ P2 | Cooperate | Defect |
|---|---|---|
| Cooperate | (1,1) | (-1,2) |
| Defect | (2,-1) | (0,0) |

We present five well-studied possible scenarios and their outcomes:

1. *One-shot game*: If the game is played once, it is always in the best interest of

[2]As we will discuss later, it need not be the exact same players. For example, overlapping generations or one long-term and many short-term players can be sufficient [38].

each player to play Defect, regardless of what the other player does. Therefore, the unique Nash Equilibrium (NE) of this game is (D, D).

2. *Finite game with known number of rounds*: Playing D in every round regardless of the other's play is the only NE. Thus, the observed outcome is again (D,D). This conclusion comes from reverse induction. The last period is exactly the one-shot game, and given (D,D) in the last period, there is no additional reason to cooperate in the second-to-last period. Therefore, the second to last period becomes equivalent to the one-shot game, and so on.

3. *Infinite game with perfectly patient players*: With an appropriate threat to punish defectors (i.e., playing D forever if one's opponent plays D) we may now have a NE with observed outcome of (C,C). So long as the each player is patient (not desiring to forgo future payoffs for the short term gain of cheating), each will continue to play C. Therefore, it's in the other's selfish interest is to continue to play C as well.

4. *Infinite game with impatient players*: This scenario can be considered a hybrid between (3) and (1). As the players become more patient, we approach the outcome in (3) and when they are less patient we approach (1). For players that are less than perfectly patient, we must compare different time series of payments. To do so, it is standard to use a *discount factor* to capture the fact that future payments are less valuable. Typically, this factor is represented by by $\delta$ ($0 \leq \delta \leq 1$), and can – for example – represent the time-value of money. Here $\delta = 1$ represents perfectly patient players whereas $\delta = 0$ represents perfectly impatient players.

5. *Finite game with unknown horizon and patient players*: This model is functionally equivalent to (4) if we view the probability of the game ending at any point in time to be a random (Bernoulli) variable. As the expected horizon increases, we approach the case of (3) and as it shortens we approach the case of (1). We can again capture this future discounting with the discount factor, $\delta$, and use similar analysis.

It is interesting to note that both cases (4) and (5) can be analyzed in the same fashion, using $\delta$ as a parameter to understand the space between the extremes.

For example, assume that Player 2 plays the following strategy:

1. Play C

2. If P1 ever plays D then play D forever.

Now look at player 1 in either (4) or (5), fixing the strategy of P2 as above. If she cooperates she receives $u_1(C,C) = 1$ forever. However, if she deviates, she obtains $u_1(D,C) = 2$ once and then $u_1(D,D) = 0$ for the rest of the game.

The sample strategy will be an equilibrium strategy if and only if the payoffs of playing the strategy are greater than or equal to the payoffs of deviating from the strategy and suffering the consequences.[3] Again, we model player's preferences over streams of payoffs by discounting the the payoffs with a decaying parameter, $\delta$. Using the above values, we can determine whether or not the sample strategy is an equilibrium strategy by comparing the payoffs to various actions.

Logically, we want to write the expression:

$$\text{Playing the strategy} \geq \text{Cheating} + \text{Suffering the consequences}$$

Formally, this can be written as:

$$\sum_{t=0}^{\infty} \delta^t u_1(C,C) \geq u_1(D,C) + \sum_{t=1}^{\infty} \delta^t u_1(D,D) \tag{2.1}$$

The left side is the discounted stream of payoffs from playing C forever. The first term on the right is the one-time payoff to cheating and the second term is the stream of payoffs that result. This simplifies:

$$\frac{u_1(C,C)}{1-\delta} \geq u_1(D,C) + \frac{\delta u_1(D,D)}{1-\delta} \tag{2.2}$$

We now put it in the more standard form:

$$u_1(C,C) \geq (1-\delta)\, u_1(D,C) + \delta u_1(D,D) \tag{2.3}$$

Substituting the values from the problem above we see that for $\delta \geq \frac{1}{2}$ the player will want to cooperate given this particular strategy.

The above analysis is quite simple but yet quite powerful and flexible. It demonstrates how a simple strategy in the context of a repeated model can qualitatively change the outcome of the game. This can be viewed a positive since it can rationalize and present a much larger set of outcomes. However, it can also be considered a weakness due to the large set of possible outcomes and thus lack of prescriptive

[3]This claim is offered without proof here. In Chapter 4, we discuss that this follows directly form the one stage deviation principle for subgame perfect equilibria.

power. However, the above example demonstrates that it is possible to analyze the equilibrium expression (e.g., Eqn (2.3)) to more firmly understand what equilibria were possible. We will see this analysis technique applied in the context of routing (Chapter 4) and overlay networks (Chapter 5).

## 2.3 Key Terms

We present some standard terms and notation that we will be using in the rest of this thesis:

- *Subgame:* A subgame is the subset of an original game beginning at a particular point and continuing to the end of the original game. Further, all (relevant) history of play is known to all players.

- *Strategy space:* A strategy space is a set of strategies that meet a set of restrictions. We use this concept to define the set of strategies that we are willing to consider for the purpose of a particular analysis.

- *Profit Function:* $\pi(p_i, p_{-i})$*:* When dealing with firms, we denote the per stage-game profit[4] using the function $\pi(.)$. In our examples, this function is thus defined differently for each game that we analyze. The parameters of $\pi(.)$ are $p_i$, the play of player $i$, and $p_{-i}$, a vector representing the play of all other players. When play of the other players is symmetric, we can write $p_{-i}$ as a single number without loss of generality.

- *Weakly dominant:* A strategy is weakly dominant if it always does at least as well as any other strategy, regardless of the strategy selected by the other players.

- *Mechanism:* A mechanism is a procedure which takes a set of inputs and produces a set of allocations and a set of required payments. For example, in the case of a first-price sealed bid auction, the mechanism allocates the good to the player with the highest bid, charges that person her bid, and charges everyone else zero.

- *Strategyproof:* A one-shot mechanism is strategyproof if truth-telling about one's private information is weakly dominant. In the repeated game, we define

[4]Since these are firm, we use the terms "payoff" or "profit" instead of utility.

a mechanism to be strategyproof if the strategy function (which takes both one's private information and the game history to date) that always plays truthfully is weakly dominant.

- *Vickrey-Clarke-Groves (VCG) Mechanisms* The Vickrey-Clarke-Groves Mechanisms are a family of mechanisms that are strategyproof. These can loosely be considered a generalization of the second-price auction. Through payments taking on a particular form, users are incentivized to be truthful about their private information. Due to its general structure and applicability, the VCG mechanisms are quite popular.

## 2.4 Notation Summary

For reference we summarize the standard notation we use in this thesis. $u(.)$ and $\pi(.)$ are used to represent utility functions and profit functions respectively. We refer to the functions as utility functions when the players are individuals and profit functions when they are firms. Typically Latin letters (e.g., $b$, $k$) are used to represent system parameters whereas Greek letters (e.g., $\alpha$, $\beta$) are used to represent game theoretic concepts such as strategies. One exception is $\delta$, which always represents the discount factor.

We use subscripts and superscripts extensively. Typically, a subscript (e.g., $b_i$) represents player $i$. The subscript $-i$ (e.g., $b_{-i}$) represents a vector of all players other than $i$. Often we consider symmetric strategies and thus can treat a term indexed by $-i$ as a single term rather than a vector without loss of generality. Typically superscripts are used to index time (e.g., $\beta_i^t$ for the move prescribed by strategy $\beta$ for player $i$ at time $t$). One exception is the superscripts $I$ and $II$ which are used in Chapter 4 to refer to the price of the 1st price and 2nd price auctions respectively as $p^I$ and $p^{II}$.

All terms and notations are also defined within the relevant text.

## 2.5 Granularity of Action Space

As discussed in the Introduction, the granularity of the action space can significantly impact the outcome in repeated games. This is important since most networked applications, communicated with bits on a wire, have an inherently discrete action space. Further, the size of this space (the number of bits allocated to the value) is

itself a parameter. Indeed, this concept will play an important role in Chapter 4. To gain insight into this phenomenon, we present a simple illustrative example here in a semi-formal fashion.

Consider a simple game where two firms are selling identical goods. Each has an infinite supply but per unit cost $c = \frac{1}{3}$. They come to market with their price $p$ to meet $N$ buyers. ($N$ is even.) If they are priced equally, each sells $\frac{N}{2}$ goods at the price offered. Otherwise, the lower priced provider receives all $N$ at the price it offered.

For example, if both offer a price of 10, they will both obtain a profit of:

$$\text{profit } = \frac{N}{2}\left(10 - \frac{1}{3}\right)$$

Assume that the firm offers prices in the standard format, with two decimal points of granularity (e.g., \$2.19, \$3.26, \$0.99). Consider a case where both firms offer – for example – \$1.00. Here, one firm could lower its price to \$0.99. This would increase profits – the increased traffic would offset the price reduction. Therefore, no firm – in the one-shot game – should be willing to offer a price of \$1.00, and this logic applies for all prices above \$1.00. Instead consider a price of \$0.34. Here, one firm could reduce price to \$0.33 – but now it is no longer making a profit. Indeed, both firms would be happy (and stable) at a higher price. Continuing this logic in a formal fashion will show that the equilibrium of the one-shot game here is \$0.36. For any price higher, the other firm will undercut and get all the market. With a lower price, the firm could increase volume, but the net profit will be less.

Assume now that each firm is restricted to offering integer prices (i.e., \$1.00, \$2.00, ...). Now the equilibrium of the one-shot game is \$2.00. The logic here is the same – a lower price increases volume but lowers profit; and a higher price will be profitably undercut.

This logic can be generalized. Let $b$ represent the minimum price change – that is $b = 0.01$ in the first example and $b = 1.0$ in the second. In general, we can solve for the equilibrium price $p^*$, which will be roughly $2b$ greater than the cost. The intuition here is that decreasing by $b$ doubles traffic but halves profit – for no gain. Since we have a discrete action space, the formal value for $p^*$ is slightly more complicated and given by:

$$p^* = \left\lfloor \frac{c + 2b}{b} \right\rfloor b = \left\lfloor \frac{1}{3b} + 2 \right\rfloor b \tag{2.4}$$

(The floor function above discretizes $c+2b$ to the appropriate level of granularity.)

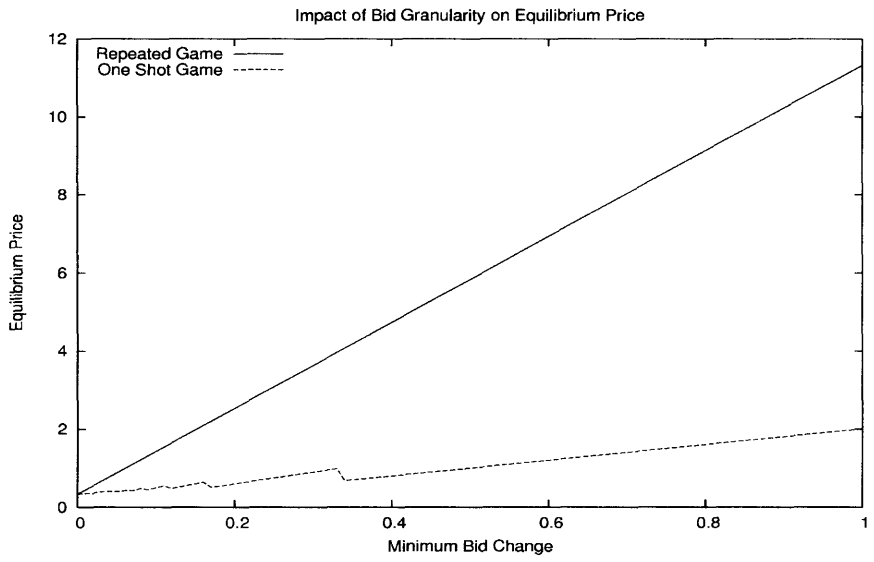



Figure 2-1: The Impact of Bid Granularity on Equilibrium Price. The impact of the bid granularity is significantly greater in the repeated game.

Let us now consider the repeated version of this game. Here the same game is played each day for all time. Let time (the day) be represented by $t$ and the first game be $t_0$. Consider a simple strategy for this time.

1. At time $t_0$, offer some price $p^*$
2. For $t > t_0$, offer the minimum price advertised by any player thus far

This strategy punishes deviations by never raising price again.[5]

To analyze the repeated game, we start with an equation similar to Eqn (2.1):

$$\sum_{t=0}^{\infty} \delta^t \frac{N}{2}\left(p - \frac{1}{3}\right) \geq N\left(p - b - \frac{1}{3}\right) + \sum_{t=1}^{\infty} \delta^t \frac{N}{2}\left(p - b - \frac{1}{3}\right) \tag{2.5}$$

Which, after simplification yields:

$$p^* = \left\lfloor \frac{c + 2b - \delta(b+c)}{1-\delta} \right\rfloor b \tag{2.6}$$

(Note that for $\delta = 0$, this simplifies to the solution for the one shot game.)

We plot Eqns (2.4) and (2.6) – for a fixed $\delta = 0.9$ – in Figure 2-1. This clearly demonstrates that *the impact of bid granularity is significantly greater in the repeated game.* In particular, the slope of the repeated game curve is $\frac{2-\delta}{1-\delta}$ whereas the slope for

[5]This strategy is subgame-perfect.

the one shot game is 2. For $1 > \delta > 0$, the former is strictly greater. In a real game, there may be several other mitigating factors, but to the extent that the repeated game effects are significant, this underlying phenomenon will still likely hold.

## 2.6 Related Work

The origins of games as a modeling tool trace as far back as 1838 and 1883 to the the works of Cournot [20] and Bertrand [7] respectively. Game theory itself is typically credited to the modern contributions of von Neumann and Morgenstern (1944) [100] and John Nash (1950) [71]. From that time, game theory has been a growing sub-field of Economics and increasingly applied to other disciplines.

One of the fields to which game theory has been applied has been Computer Science and in particular networking. Game theorists have long used computers for simulation of various games and strategies. Notable examples include the Axelrod experiments in which various strategies for the Prisoners' Dilemma were simulated [82]. More recently, due in part to the reasons presented in the Introduction, many researchers have applied game theory to better understand and/or design networked systems. Routing has been one popular area of study, with Kelly *et al*'s work on congestion pricing [57], Nisan and Ronen's application of the VCG mechanism to routing [74], and recent work by several research researchers on efficiency loss in congested networks with distributed selfish control [53, 85]. For general networked and distributed systems, differing subsets of Feigenbaum, Papadimitriou, Sami, and Shenker present a series of results on distributed algorithmic mechanism design [32, 34], some of which are summarized in [35].

We examine three lines of literature below. The first is work from Economics on repeated games in general conditions – important as it builds confidence that repeated game models are robust to practical environments. Then we examine the work of Dellarocas on reputations, which is only tangentially related in substance, but is very similar in approach to this thesis. Finally, we consider games of network formation, very related in spirit to the problems considered in this thesis, particularly the overlay problem of Chapter 5. In addition to this section, the individual chapters of this thesis present the work related and relevant to the particular problem(s) considered.

### 2.6.1 Repeated Games Under General Conditions

While the repeated Prisoners' Dilemma analysis above focuses on a narrow scenario, such repeated game logic has been shown to be applicable to a variety of contexts and robust to a variety of assumptions regarding the game – in theory and practice. The existence of these non-trivial repeated equilibria is often called the "folk theorem" and there are a number of folk theorem results, each showing similar results under different (or more general) conditions. Most folk theorem are of the form that – under particular assumptions – any strictly individually rational payoff (payoff better than not participating in the game at all) of the stage-game is a feasible average payoff of the repeated game. Typically these results require a sufficiently high $\delta$ – that is they hold for $delta \geq \bar{\delta}$, for some $\bar{\delta}$. What this means in practice is that a large number of alternative outcomes are possible in the repeated game, conditioned on properties of the underlying game. In general, folk theorems have been shown under a wide-variety of weakening assumptions such as imperfect information [39] [44], players of different horizons [38], and even anonymous random matching [27]. While our research does not rely on some of the more powerful mathematic analysis of these results, they are important as they demonstrate how the concepts and tools used in this thesis can be applied to a much more general class of problems.

Even more important, we see the effects of repeated interactions in practice quite often. One broad class of examples is that of oligopolies, such as cartels, which draw their strength from the repeated interactions. Here a small number of firms control a market and through price and/or quantity controls, maintain a price higher than the case of the one-shot (Bertrand) competition. In Chapter 4, we will discuss how the "Price Match Guarantees" offered by many retailers that we are familiar with also relate to repeated games. These sorts of behavior often rely on the ability of firms to implicitly coordinate and communicate, since in general market collusion is hard and perhaps even illegal. Such implicit communication patterns have been observed in real markets. In one example, firms in a telecommunications auction signaled their interest (or lack-thereof) in certain licenses through careful constructed prices [58, 59] that limited their future moves. In another telecommunications auction, a firm encoded (in the digits) information about the territories in which it is interested [22] as a way of signaling a potential compromise to another firm.

### 2.6.2 Dellarocas' Investigation of Reputation Mechanisms

The body of work most similar to this thesis in approach is the research by Dellarocas use of repeated games to study reputation mechanisms [25, 24]. Reputation mechanisms are a potentially useful tool in a variety of contexts, but pose several interesting implementation questions. (For more background, refer to [87].) For example, there are parameters such as how often the reputation should be updated, how to calculate the reputation score, and even how many states the score should have. Dellarocas' research examines these questions in the context of an auction site such as EBay, where each seller has a reputation. To do so, he models the interaction between a monopolist seller and a set of random buyers as a repeated game. This interaction is mediated by a reputation mechanism, which is a function of the particular parameters he is examining (such as the ones mentioned above). He uses repeated game analysis to derive practical, and sometimes counter-intuitive, results which can be employed to develop better mechanisms.

As such, Dellarocas' work has several key parallels to this thesis. The problem examined cannot be readily solved in the one-shot game; a repeated game is a natural and correct model for the interaction; and the use of repeated games provides practical insight. His work differs from this thesis in that he focuses only on the problem of reputation mechanisms. Further, while the problem is cast in an online setting, the analysis in no way relies on that property, nor are the network or protocols involved in the problem. Nonetheless, the strong analytical parallels strengthen our belief that repeated games are an appropriate and practically useful tool.

### 2.6.3 Network Formation Games

Another related set of literature is that on network formation games. Here the games are between players, represented by nodes, who form edges to other nodes based on some objective function, and in general can encompass a large class of problems. There is an extensive treatment of these problems in the Economic literature, in particular Matthew Jackson has investigated many such problems and presents an overview of some of the work in [51]. In the Computer Science literature, there has been some work motivating and characterizing the shape of topologies that result from selfishly motivated network formation processes [29, 16, 19, 93]. In particular, Fabrikant *et al* [29] motivate their work by considering each agent to be an independent network. Christin and Chuang look instead at individual users and propose a cost model for participating in an overlay topology [12]. They then use this to

evaluate several overlay architectures.

The work in this thesis shares inspiration with much of this work. The notion of selfishly motivated agents affecting the network topology and performance is a theme shared through this thesis. Chapters 3 and 4, like [29], concern themselves with selfishly motivated networks. Chapter 5, like Christin and Chuang (and other related work discussed in the Chapter 5) recognize the incentives of individual users in the overlays and the cost associated with the load induced on each node. Further, Chapter 5 examines the impact of different overlay topologies on this dynamic.

There are however several key distinctions between our work and these papers. The most striking difference is that all of these models consider the one-shot game. The precise problems considered themselves are also different. For example, [29, 16] consider hop-count as the utility function whereas our research considers profit as the objective function of the networks. Similarly, [12] examines a variety of file-sharing peer-to-peer overlays, while our work in Chapter 5 considers application-layer multicast. Further, the games of networks assume that the physical topology is mutable, in our games it is static. This has an impact not only on the particulars of the problems, but also on the type of results obtained.

# Chapter 3

# Benefits and Feasibility of Incentive Based Routing

Routing on the Internet today is as much about money as it is about traffic. The business relationships of an ISP largely dictate its routing policy and drive the work of its engineers. In today's routing mechanism, this leads to a number of well-known pathologies.

This structure is further challenged by the emergence of user-directed routing, which turns the problem of routing into a multi-party repeated game. This chapter explores these challenges and argues that the repeated model motivates the *introduction of explicit incentives (prices) into the routing systems used on the Internet.* We argue that doing so addresses limitations of the current system that are significant today and will only be exacerbated by user-directed routing. To support this claim, we describe the benefits and properties of incentive-based routing frameworks and demonstrate how such frameworks can be applied to a number of routing architectures, including BGP.

## 3.1 Introduction

Scalability and decentralization are desirable attributes for Internet-scale inter-domain routing systems. Today, this is achieved through a distributed system in which communication between networks happens pair-wise. It is therefore vital that networks be able to make decisions myopically – based on the local information that they have at hand – and that these decisions are stable. A model of inter-domain routing pro-

posed by Gao and Rexford [42, 41] argues that under a certain set of assumptions the structure of inter-ISP business relationships induces an equilibrium where each Autonomous System (AS) acts based solely on its contracts but "where no AS would change their routes," [42] a notion they call stability. This notion of stability is the equilibrium of a game of incomplete information where each player's (private) type is its set of business relationships. To obtain their result, Gao and Rexford implicitly assume that traffic patterns are under the control of ISP and that this control allows inter-ISP business relationships to be relatively stable.

We argue that these assumptions are both difficult to maintain in the current Internet and under increasing pressure from emerging *user-directed routing* technologies. We suggest that user-directed routing exacerbates underlying problems in the current system by creating a significant misalignment of incentives. As a result, ISPs are forced to appreciate the full-complexity of the *repeated game* including predicting future traffic patterns and guessing how others will react. These factors place ISPs and their customers (both individuals and other ISPs) in direct conflict. In the terminology of Clark *et al*, this is a "tussle" [17].

In this chapter we focus on the use of repeated games as a tool to better understand this problem. First we explain that the introduction of user-directed routing has changed Internet routing into a meaningfully repeated game. By analyzing the Gao and Rexford model in light of the repeated game, we are able to explain the tension observed in practice and some otherwise inexplicable behavior. In this chapter we do not propose a new solution *per se*, neither do we present complete analysis of a new class of model. Rather, we motivate the tighter coupling between routing and pricing mechanisms and present some sample architectures in which this can be done readily, both for current (BGP-based) routing systems as well as potential future systems. In the next chapter, we examine the question of designing the actual protocol and analyze some of the important subtleties and their relationship with this underlying incentive problem.

The contributions of this chapter are therefore:

- Demonstration that user-directed routing has transformed routing to the point that the repeated game model is the most appropriate.

- Observation that when viewed via the lens of a repeated game model, inter-domain routing may lack the stability that is desirable from a system-design perspective.

- Motivation for the design decision to couple routing and pricing mechanisms. We consider one set of the practical subtleties of this problem in Chapter 4.

## 3.2 The Nature of Today's Internet Routing

The routing fabric of the Internet has a large number of players and technologies. In this section, we briefly explain some of this technology for those who may not be familiar with it. (Those interested in more information may consider [79, 98] on networking and routing or [54] on BGP.) We then provide perspective on how routing is *actually* done in practice on today's Internet, arguing that this complex machinery is largely driven by a much simpler factor: money.

### 3.2.1 Inter-Domain Routing Overview

The Internet is comprised of a number of independent networks, including commercial networks (e.g., AT&T, SBC), universities, and government networks. From a technical perspective each is considered an Autonomous Systems (AS) and assigned an AS number. Typically, the AS is also allocated a set of IP addresses and is responsible for all routing done within the particular network.

To provide connectivity between networks, the ASes inform each other of paths to IP addresses in different networks. This process of information sharing and the subsequent directing of traffic is called routing. Inter-domain routing on the Internet today is done via the Border Gateway Protocol (BGP). BGP allows a network to tell the other networks to which it has physical connections about the destinations within its network. Similarly, BGP allows networks to relay this information along to other networks. For performance and expressive power, BGP permits each AS to associate additional information with each advertisement. The most basic information is the AS path to the destination. This can be used to prevent loops in routing and to allow an AS to select the shortest path to a destination given multiple potential paths.

A key property of BGP is that each AS makes its own local decisions. In particular, when one AS hands a packet off to another AS, where that packet goes next (the "next-hop") is solely at the discretion of the AS that currently now has the packet. Similarly, an AS can filter the information that it shares. For example, it can choose not to advertise routes that it knows about. Further, for most information, there is not even a requirement that the AS report its information truthfully. For example, one common technique for load-balancing and other traffic engineering is AS path

length "padding." Here, one ISP will lengthen the reported path length to a given destination by inserting its own AS number multiple times. This can cause another AS to prefer one route (the one with the shorter length) over another (the the longer AS path length) and is a crude (yet prevalent) means of traffic engineering.

### 3.2.2 The *Reality* of Today's Internet Routing

In theory, today's Internet routing system allows ISPs to operate with arbitrarily complex and independent policies. In practice, however, ISP policy is normally quite simple and driven by one motivating factor – money.[1] The policy of these ISPs can be characterized as an exercise in cost minimization: *"Given that I had to accept this packet and now must forward it, what is the cheapest route on which to do so?"*[2]

These decisions are driven by the inter-network business relationships, which fall into two broad classes. The first is the customer-provider relationship. Here one network pays the other for the traffic that passes over the link(s) between these two networks. The second major class of business relationships is peering. Peering relationships are typically formed by larger ISPs, where both agree that for traffic on routes advertised to each other, no payment will be required. While there are exceptions, peering relationships are typically formed when the traffic exchanged (and thus the potential money exchanged) is roughly equal.

## 3.3 Downsides to the Current Model

While the current model of inter-domain routing has supported the Internet thus far, it has several significant and well-recognized downsides. At the heart of many of these problems is that ISP economics and users' desires are the fundamental quantities of the system, but are not represented in the routing protocols themselves. Instead, the ISPs must fall back on imprecise and indirect BGP techniques to convey and act upon incentives. As a result, both users and ISPs suffer in many ways:

1. **Sub-optimal Routing** Because the AS is the player who makes the routing decisions on behalf of the users, there exists the potential for a significant *moral*

[1] In this discussion we ignore non-commercial networks, such as government networks, where factors such as privacy may be tantamount. However, even there the problem is not very different as it is simply a constrained optimization problem, selecting the cheapest route over a series of permitted links.

[2] We ignore the strategy of simply dropping large amounts of traffic, as ultimately no one will do business with an ISP who adopts such a strategy.

*hazard.* That is, given a fixed amount of traffic to route, the AS can, will, and (as a profit-maximizing firm) in many cases *should* make decisions which will decrease (or perhaps even minimize) its cost, at the expense of poorer service to the user.[3]

A well-understood example of this phenomenon is "hot potato routing" [99], where an ISP hands packets off to peer ISPs as rapidly as possible. This occurs because the peer appears free, a high incentive to use a possibly sub-optimal route. A second example is traffic routed to stabilize the ratio at a peering point – here the mis-incentive is the possible loss of the peering relationship if traffic becomes unbalanced. Sub-optimal routes due to such mis-incentives can cause decreased performance for end-to-end flows and BGP itself. While economics is not the only reason, it has been observed that 30-55% of the paths on the Internet are sub-optimally routed [90]. Still other examples include misrepresenting routes (e.g., path length padding) or simply not advertising an existing route at all.

2. **The Costs of Inter-Domain Traffic Engineering** Another cost of the current system is the work and risk associated with traffic engineering. As IP service commoditizes and profit margins decrease, ISPs and researchers have begun paying closer attention to costs of implementing policy. This process is often complex and/or manual [30] due in part to the fact that there are few ways to cleanly implement policies. For example, consider the use of AS path length padding to direct a fraction of traffic over a particular link. Not only is this process inherently manual (requiring a human operator to make such a change) but its success, in many cases, is dependent on properties of the traffic and worse the actions and reactions of other networks. As such, successful implementation requires periodic monitoring and potentially (more manual) updating.

   Taken together, this is costly in two ways. First, complex manual process is a financial burden to the ISPs – the process of cost-minimization through traffic engineering is itself costly. Second, the complicated and human process can easily introduce significant errors into the routing system [30]. Simplifying and automating this particular process can both reduce costs and improve the level of service.

3. **Instability of Peering Relationships** Because peering contracts provide

[3]We discuss in the next section that examples such as CDNs validate that these problems are both significant to users and addressable.

tremendous cost savings, ISPs exhibit *adverse selection* to obtain and maintain them. It becomes rational to take varied questionable steps to obtain and/or maintain these relationships. One approach taken is to make side-deals with other AS or entities to force a certain amount of traffic through a given inter-connection with a third party. This extra traffic is designed to bring the ratio of traffic at the inter-connect to a given level. Another approach is to host systems which are large traffic sources or sinks (e.g., SourceForge [94]) as a means of balancing ingress and egress traffic. A key reason that so much effort is invested in this process is that there is no graceful transition between the relationship of peer and that of customer-provider. This problem was most painfully obvious during the C&W/PSI blackout of 2001 [72] where an argument over a peering relationship with no clear resolution affected thousands of users and corporations.

4. **Lack of Price Discrimination** In addition to increasing costs for the networks, the current inflexible scheme decreases potential ISP revenue. Unlike many more established systems (telephone, postal, airlines) most Internet pricing is based on a single rate applied to all usage. This pricing has the merits of simplicity and small-customer acceptance, but it is well known to reduce economic efficiency. As ISPs focus more on rates of financial return, the ability to discriminate on customer willingness to pay becomes a more important tool. Indeed, finer-grain differentiation has emerged in the maturation process of other networks such as transportation [75]. Indeed, the industry is starting to move toward more sophisticated models. For example, some networks offer different on-net vs. off-net pricing and one router vendor recently began offering some ability to do per-destination accounting [49]. However, this practice is still nascent, limited, and forming in an ad-hoc fashion.

We observe that to the extent each of these problems is addressed today, it is done without explicit protocol visibility of ISP objectives and incentives. In some cases the objective is undefined; there is no standard inter-ISP quality metric. In other cases the incentive is defined but outside the reach of the decision-making protocol; inter-ISP financial incentives are defined by paper contracts, not the routing system.

Absent this information, ASes are forced to resort to complicated and imprecise tools (e.g., AS path padding, BGP communities, or any of the other sundry BGP options). Thus, no AS understands how its decisions impact its neighbors and users, nor can it communicate the cost of such decisions in a way to be effectively compensated

for them. Devoid of a more efficient means of achieving an efficient equilibrium, the AS is often left to make arbitrary decisions, and to implement them in a costly (manual) process of trial and error (e.g, repeated manual consultation of MRTG graphs [76]). This process is costly, complicated, and inefficient for all parties involved.

From these observations we draw two conclusions:

**Conclusion 1.** *Maintaining business relationships and control over traffic in the Internet today is costly to ISPs.*

**Conclusion 2.** *Much of this cost and complexity stems from the fact that the financial incentives are not explicitly communicated in the tools and protocols that are used.*

In other words, the apparent simplicity of today's model is specious. The complexity we have removed from BGP has only created work and complexity elsewhere in the system. Further, the added work and complexity not only decreases the value of the network but also increases its cost structure.

## 3.4 User-Directed Routing

To address some of the short-comings of today's routing architecture, numerous methods to provide some element of **user choice** in routing have been proposed, built, and (sometimes) deployed. These systems are motivated by the complexity and inexact nature of the routing protocols and tools available today and inimical side-effects discussed. In many cases, these approaches are able to circumvent some of these factors. We refer in this chapter to *user-directed routing technologies* when focusing on the broad principle rather than any particular implementation.

Today's most common examples of user-based routing are overlay networks. In industry [3] and academia [89, 5], overlay technologies have been used to increase the reliability and performance of Internet flows. Overlay networks operate by placing overlay nodes at various places throughout the network. Traffic, once directed to the overlay network, travels to the destination via the nodes in the overlay. While the traffic travels along the standard BGP path between the overlay nodes, the overall path taken by the traffic can be very different than the path prescribed by BGP. This is depicted in Figure 3-1.

Overlays exploit two fundamental facts of today's Internet routing. First BGP has no true notion of quality-of-service (QoS) and certainly no notion of end-to-end QoS. In particular, BGP has no visibility into packet loss nor understand the requirements

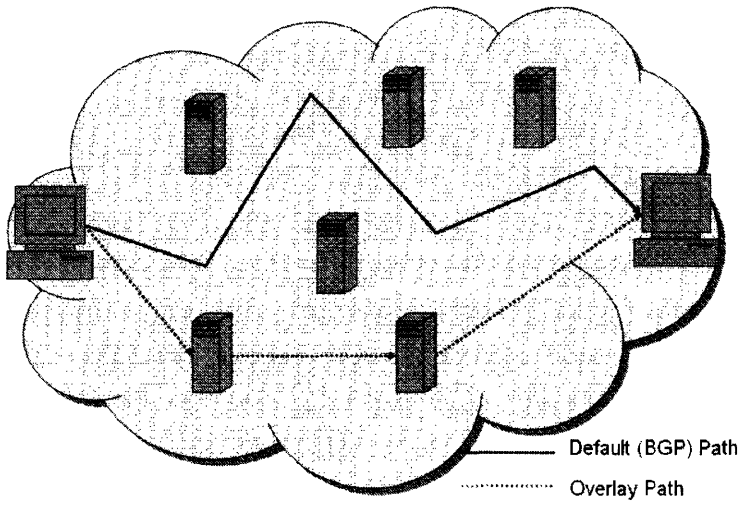



Figure 3-1: Overlay Routing: The flow does not take the default BGP path through the network. Instead, it travels between servers in the overlay. While the path between servers is dictated by BGP, the overall path is not.

of corresponding traffic flows. Further, what controls do exist largely do not work across ISPs. The few metrics BGP can use (e.g., AS path length) can – and often are – corrupted by economic incentives. Even worse, many routes are not even advertised to other networks for economic reasons. Second, overlays address the moral hazard problem. By distributing the choice to the end-user, the only agent who is properly incented to pick the route that appears most appropriate for the particular task at hand, they shift the balance of control.

Overlays are not the only form of user-directed routing. Overlays are constrained in that they require the additional overlay nodes and through which the traffic must flow. To address these downsides, there have been a variety of proposals (e.g. [18] [103] [81]) that provide for even greater user control in route selection by placing the user in charge of routing decisions deeper within the network. (These are loosely called source-based routing.) A third area of user-directed routing can be found in peer-to-peer networks. Here the routing is much less specific from a network perspective and is instead driven by the availability and location of the relevant content. When a given piece of content is found in multiple locations, the application or P2P algorithm then makes the decision about how the content will be obtained.

Although these user-directed routing proposals contain both common elements and sweeping differences, a key detail is that many – most notably overlays – can be created without the support of ISPs. This, for example, is a critical difference from IP Multicast, another technology that posed an economic threat to ISPs. Based on

this, we therefore make two observations:

- *User Choice increases the fluidity of traffic patterns by several orders of magnitude.* A traffic source (e.g., a large company or web site) might for example change its ISP once a year. By contrast, an overlay network can for example shift the site's traffic on the order of every 5 minutes. Under this hypothetical, we have a factor of 100,000 change – five orders of magnitude. Of course, most overlays can adjust a significant fraction of traffic even faster than that. Moreover, these changes need not be stable – traffic could be rapidly shifted back and forth. These shifts significantly perturb capacity planning and peering relationships.

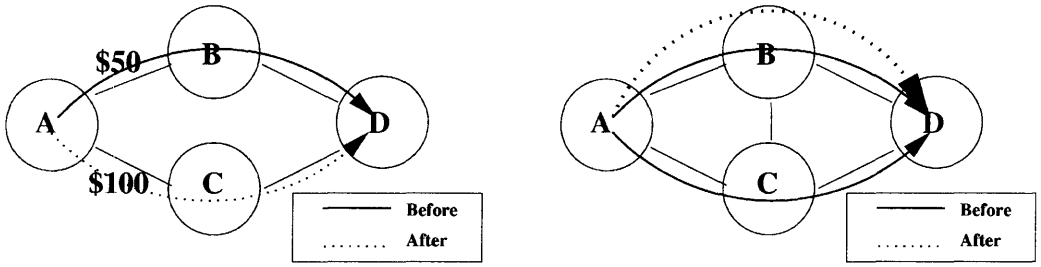
- *It is not acceptable to assume away the effects of user-directed technologies, deployed with or without the cooperation of ISPs.* Akamai, a Content Delivery Network, alone today carries about 10-20 percent of the web's traffic [2] and for some networks, P2P traffic can be the largest source of traffic [10]. We must examine the impact of user-directed routing on the current incentive structure, and should go beyond this to examine what framework is best suited to support user-directed routing's growth.

These two observations imply a meaningful change to the nature of routing on the Internet. Routing is no longer simply at the control of a given ISP or the bilateral communication between ISPs but truly a *multi-player game* with the users as meaningful participants. Further, a corollary of the first observation is that this game is played out continually over time. Based on the flexibility of the given user-directed routing technology, this game can be played out tens or hundreds of times per day. As such, this clearly motivates the consideration of a *repeated-game model of routing.*

## 3.5 The Core Tussle

From the simple explanations of Internet economics and user directed routing, it is immediately clear that we are facing a tussle of significant magnitude at the very core of the Internet. On one side, users are demanding choice and the other ISPs are trying to maintain fragile business relationships. In particular we state the following:

**Conclusion 3.** *The already fragile set of business relationships that underly the routing fabric of the Internet will be challenged by user demands, in the form of user-*




(a) Priced Service (b) Peered Networks

Figure 3-2: Routing Examples

*directed routing technologies. Without some means of rationalizing the economic interests of the ISPs with the desires of the users, the Internet will suffer from decreased quality and increased cost.*

### 3.5.1 Examples

To crystallize the problems and provide reference points, we present three simple examples.

**Example 1**: In Fig. 3-2a, AS A is a customer of both B and C to reach a set of destinations D. The price for B is \$50/Mbps but \$100 for C; thus A uses B. Now assume that a significant fraction of A's users wish to travel through C to reach D. A obviously has significant disincentive to allow this. In this situation, it is likely that A will block any form of user directed routing at all, if it has the ability to do so. If it does not have this ability, then it is clear that A will suffer.

Neither outcome is a good one.

**Example 2**: Example 1 relied on price differences to create the tension. However, many networks are peered (settlement free). Consider the similar example of Fig. 3-2b. Here A, B, and C are peers, with peering traffic ratios close to 1.0. Let us now assume that A's users can and wish to direct their traffic through B.

After some time A, B, and C will notice that their peering ratios are now significantly out of balance. If A is the smaller ISP it may now be forced to pay B for its future traffic – or worse it may be forced to pay *both* B and C. Knowing this, A will attempt to redistribute the portion of its traffic that is not user-directed. This solution comes with operational costs and causes traffic to be sub-optimally routed. Beyond this, a feedback loop is created; the poor routing may incent users to depend

further on user-directed routing, worsening the problem.

Again, the outcome is poor for A, A's users, or both.

**Example 3**: This third example is slightly more complex, but is designed to emphasize an insidious mode of cheating which violates one of the rules proposed by Gao and Rexford. Consider the network depicted in Fig. 3-3. In the figure arrows represent the initial flow of money. We are going to see that the mid-sized ISP, M, is going to try to force traffic through its provider, L, instead of its customer, S, in an attempt to preserve the customer-provider relationship with S.

In particular, assume we have the following scenario to start:

- y is a customer of S
- S is a customer of M
- M is a customer of L
- y is also a customer of L
- P has connectivity through M and L. P could be a customer of both or a peer of both.
- Traffic from P flows through L to y.

Again, the arrows in Fig. 3-3 depict these relationships.

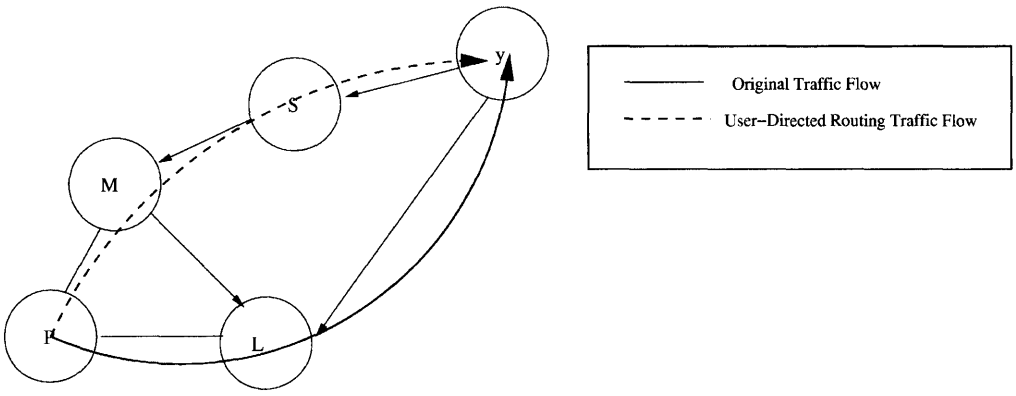



Figure 3-3: A Potential Problem: Traffic is flowing to y via L. P directs this traffic through M instead. Since impacts M's business relationship with S, M may try to force the traffic via L instead.

Let us now assume that P's downstreams, through use of a user-directed routing system determine that they want the traffic to flow to y via M. Normally we would

expect traffic from M to y to flow through S. But if the relevant traffic is significant, it could equalize the traffic ratio between S and M. Because the business relationship between two networks is typically based on the traffic ratio, bringing this ratio in balance could cause S to want to peer to M. M, knowing this, could decide to try to avoid the problem by instead sending some traffic, destined to y via L. While this may cost M some money, it may in the end be the profit-maximizing decision for M. This re-shifting of traffic essentially foils the intentions of the user-directed routing system, creating a further tension.

### 3.5.2 Analysis

To better understand these examples, we turn to a formal model of the inter-business routing dynamics. As discussed, a particularly descriptive and positive model of routing in the current literature is the Gao and Rexford model. Gao and Rexford's model states that if networks route traffic via customers before peers and peers before providers, then we have a stable paths assignment. However, their model does not adequately explain the above scenarios, particularly Example 3. When M sends traffic destined to y via L, we have an example of a customer routing traffic through a provider instead of a customer.

To better understand this tension, consider a simplification of Gao and Rexford model, depicted in Fig. 3-4:

1. Commercial relationships are formed
2. Based on (1), traffic policies are designed
3. Traffic flows in accordance with (2)

When we view a particular snapshot of time, their model is still reasonable, even in the face of user-directed routing. This is because the flows, while directed by the higher level system, still travel over BGP. If the user-directed paths were fixed, this would not be a problem.

This analysis does *not* hold when the problem is played over time – that is, as a repeated game. First, the overlay removes the direct causality between steps (2) and (3). Second, step (1) in the model assumes *exogenous* business relationships. In practice, however, these relationships are based on the traffic levels themselves– they are endogenous. Consequently, there is a feedback loop, depicted in Fig. 3-4, that must be considered. If an overlay controls sufficient traffic, it can (as seen in

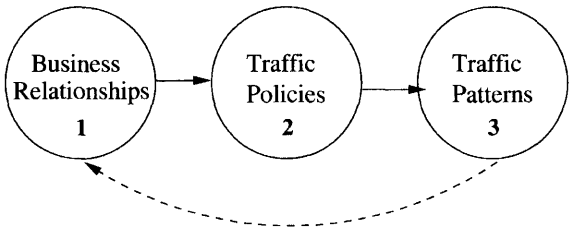



Figure 3-4: A Representation of the Gao and Rexford Model with the Implicit Feedback Loop Not in Their Model

the examples) cause an AS to regret its past decisions made – even the decision of customers over peers (precisely what is demonstrated in Example 3). Thus, the repeated version of the Stable Paths Problem, where contracts are based on traffic patterns, is not necessarily stable with user-directed routing.

The contrast in our conclusions arises due to differences between our assumptions and those of Gao and Rexford. In particular we assume a) that user-directed routing enables traffic patterns to change with significantly greater magnitude and fluidity, and b) the existence of a feedback link between (3) and (1). In reality, this link exists today, but operates at the (currently longer) timescale of traffic mobility.

The effect of our more complete assumptions is that a model of current routing systems with user-directed routing cannot be shown to necessarily satisfy the stability property identified by Gao and Rexford over time. This is unfortunate, because, as discussed earlier, the stability property leads to several characteristics valuable for Internet routing.

Although we do not prove this strong version of stability for our model, it is not the case that the Internet must therefore plunge into massive instability as user directing routing becomes widespread. We suggest that with appropriate mechanisms and models for these conditions, equally strong notions of stability can be defined for overlay-augmented systems. This thesis does not present a full design or defense for such an architecture. Instead in this chapter we examine the high level principles desirable for such an architecture and in the next chapter, we examine some of the practical implementation details of the protocol.

### 3.5.3 Potential Outcomes

Given the above structure and examples, it is not hard to imagine the set of possible outcomes over time:

- **ASes "Win"** ISPs may be successful in preventing the realization of overlays

and the establishment of any other user-directed scheme. Here user-quality will suffer, these preventive practices will create cost for ISPs, a lot of research and development effort will have gone to naught, and the benefits will go unrealized.

- **The Users "Win"** Here (as is the case today) overlay technologies allow end-users complete flexibility in picking their routes without any economic considerations. As the prevalence of overlays increases, ISP profitability and the stability of inter-ISP relationships will be significantly degraded, in turn affecting end-users. Again, this is not desirable.

- **A New Hybrid Solution Emerges** This is the desirable outcome, but also the most nebulous. In particular, it must have the property of allowing some user choice while finding a means to appropriately compensate the ISPs for the decisions made.

We observe that unlike the undesirable solutions at either endpoint of the spectrum, "the" hybrid solution is in fact a range of possible solutions, with different solutions emerging over time and different solutions being appropriate at different points in the network. Further, we have discussed several times that the root cause of these problems is that the incentives in the system are implicit, not explicit. Based on this observation, we argue that the most effective possible path forward is to make the incentives implicit in today's model *explicit* in the routing information dissemination fabric, and allow the most appropriate hybrids to emerge as and when appropriate.

**Conclusion 4.** *Introducing incentives, represented by prices, natively into the routing fabric will allow us to both resolve the conflict between ISPs and user-choice routing, and to address significant practical problems with BGP-based routing as it exists today.*

## 3.6 Resolving the Tussle

We turn now to the characteristics of the solution called for in Conclusion 4. We first consider properties desirable for any such solution and then sketch several possible implementation strategies. Our core argument is that **the ability to express pricing information regarding routes should be incorporated into the routing fabric of the Internet.**

There are three key points in the above statement. First, we believe that routes are the right economic good. By this we mean that networks sell and users buy routes to a destination, as opposed to for example, a certain level of quality. Second,

we propose the ability, not the requirement for pricing. Indeed, this is not better facilitates an incremental deployment, but also helps to address concerns regarding stability and overhead of deployment versus benefit. Third, we argue that pricing should be in the routing fabric, not necessarily the routing protocol per se. For example, in the case of BGP, while pricing information could be placed in BGP there may be reasons not to do so. Thus it could be placed in a separate control channel (i.e., a different protocol). Therefore, our point is *not* to argue for pricing in the same protocol messages *per se* but rather in the overall system. In particular, in Chapter 4 we examine several important requirements for the protocol containing pricing information, some of which may be inconsistent with simply placing it in the pre-existing routing protocol.

### 3.6.1 An Ideal Framework

We examine the properties of an ideal implementation framework here, and then later apply them to various routing architectures. The questions below are not exhaustive nor requirements; in fact we will see that tradeoffs exist. However, they provide us with guidance and metrics.

1. *"What is the good to be priced?"* Several properties are desirable. The good should be unambiguous and easily audited, to increase the likelihood of successful transactions and minimize overhead. Its definition should be directly relevant to both the end-user's utility function and/or the ASes cost, so that one or both parties can easily reason about it. (This tradeoff will be significant later.)

2. *"How and when should the information regarding the good and the prices be conveyed?"* Great variety is possible, from once a year through written contracts to every second in a routing message. We identify several guiding principles. First, the time frame should be sufficiently short that the underlying economics and incentives are stable within a given period – when an entity publishes its information, it should not worry that significant changes will cause it to regret its decisions *ex-post*. Secondly, the time frame should be sufficiently large that the system can achieve an appropriate level of stability. Third, for reasons of fate-sharing, consistency, and efficiency it is advantageous if market information about the goods is exchanged in the same framework used to convey other (e.g., technical) information.

3. *"Who are the users?"*. Thus far we have spoken of ASes and end-users. However, there is a continuum from individual users to large end-users (e.g., corporations or universities) to small ISPs to large ISPs. Furthermore, there are other players, such as 3rd party overlays. Given this melange of entities, it is unclear where different incentive-response mechanisms should be placed. In answering this question, we offer the following properties. First, the user must have sufficient information to make a decision. Second, the supplier must have sufficient ability to implement the decision. Third, the benefit to the user of being able to make the decision must outweigh the cost and/or uncertainty of having to make that decision. Fourth, the benefit to the supplier of enabling the decision must outweigh the cost and/or uncertainty of facilitating such a decision. Based on these properties, it is clear that – different hybrid models, in the language above – may exist in different locations in the system.

### 3.6.2 Application to Routing Frameworks

We turn to the question of implementing these goals in a routing framework. We look at three frameworks – BGP as it exists today, a model of complete user/source-based routing, and a model in which overlay networks interact with a route discovery protocol. In each we see that we must provide slightly different answers to the above questions.

There are two common themes in our instantiations of our principles. First, we use the notion of *routes* to represent the good in the system. Routes are easily auditable, directly implementable by the AS, and clearly tied to the AS'es cost structure. An alternative would be a good tied to a metric of quality, along the lines of DiffServ [9]. There are several reason why we select routes as our good. First, it does not require the explicit definition of quality classes. Second, it is both easily auditable and easily implemented by the ISP, obviating the need for complicated Service Level Agreements (SLAs). Third, it maps directly to the ISP's cost structure. It is important to note that while routes are the underlying good, there may in practice be various levels of equivalence classes or other groupings of routes.

The second commonality in these instantiations is our answer to when and how. Cost structures are constantly changing as contracts are renewed and the underlying topology changes. An exchange that is on the timescale of much more than days will likely not facilitate stability. All of our proposals have granularity on the order of at most hours.

These proposals are *not* intended to be complete solutions, particularly since they do not incorporate many important challenges and subtleties. Instead these solutions serve as proofs-of-concepts and motivate some important research questions. In particular, the frequency of price updates is an important and interesting issue. In the sections below, we propose sample answers but investigate this and other questions in significant formal depth in Chapter 4.

### A Next Step on BGP

The first framework we consider is BGP. One approach to incorporating prices in a system running BGP would be to create a separate protocol to run alongside BGP and inform the routers and ISPs of the pricing information. Below we instead show how BGP could be modified, if so desired, to incorporate the prices into the route advertisements. This outline could also serve as the outline for a separate protocol.

- Every inter-domain BGP route advertisement will carry with it an associated price representing a per-Gigabyte (GB) transfered price. As a matter of practice, these prices can be changed only once per time-period (e.g., one hour).
- The business relationships continue to be pair-wise between ASes, with charges now based (in part) on these prices.
- ASes can incorporate this information into their routing decisions and perhaps route solely on these prices.
- ASes can provide this information to any overlay system operating in its network.

We note several relevant implementation details. As discussed, we use routes as our logical good, but for compatibility with BGP, we represent routes as destinations since BGP enforces a one-to-one mapping at the inter-domain level. Leveraging BGP also facilitates the exchange of information without a new protocol. This is true both for inter-ISP relationships and between and ISP and its major customers, as today many ISPs maintain BGP sessions with commercial overlay providers. We also note that the per GB pricing is consistent with average-usage billing, a popular billing methodology today. Together, we conclude that such a scheme could be readily implemented by ISPs and could be deployed incrementally at the granularity of routes.[4]

[4]One subtle downside is that this could potentially cause deaggregation.

### Source-Based Routing

Next, we assume a framework where a source routing protocol is used to decide among different routes. To apply an incentive scheme we propose the following:

- Every AS associates a price with each border ingress/egress pair. As before, these prices could be in the form of a price per volume of transfer (e.g., $ per GB).
- These prices can be updated on the order of minutes.
- All information on routes and associated prices are distributed throughout the network within the routing protocol.
- Each user selects the path that maximizes her utility, given the observed quality and price of each route.
- The ASes along the path obey the requested path.
- ISPs are compensated for the use of their routes.[5]

The primary difference between the source-based and BGP-based schemes is the significant increase in information and flexibility provided to the user by the assumed protocol. Since we are not worried about the convergence of some underlying routing protocol, we can increase the frequency of price updates. Despite these differences, we again see that given the particular routing framework, we are able to infuse an incentive framework with minimal alterations.

### An Overlay Controlled Environment

Lastly, we consider an in-between and perhaps more likely reality, where overlay networks and ISPs work together to provide efficiency and scalability. The first part of the routing mechanism is a system in which path existence and pricing information is propagated through the network at some relatively low frequency. Like BGP, paths are built up AS by AS. Unlike BGP, multiple paths can be advertised and changes in link status do not necessitate a corresponding advertisement. This is because the second part of the mechanism is an overlay-based technology that chooses the optimal

[5]The means by which the ISPs are compensated is critical to the success of the implementation. Two possibilities are that the user pay each ISP along the path or that payments are accumulated pair-wise. While the latter (how the Internet works today) is more likely to be tractable, both – and permutations thereof – are permitted.

route based on the set of paths available, their relative financial cost, and their relative quality.

Here we see the following:

- Every AS advertisement also includes a price.
- This route information is updated on the order of hours.
- Overlays, based on the information at hand and users' desires make the appropriate decisions.
- Overlays may exist as separate entities (e.g., a Content Delivery Network (CDN)) and have flexible relationships with end-users

## 3.7 Results

Based on the above applications of the incentive scheme we now examine the question *"Is it worth it?"* In particular, is the main problem of the tussle and its impact on stability truly addressed? Second, does such an architecture address the problem in a way that is not excessively burdensome to the players or the market?

First we examine our impact on the tussle itself. Our schemes have made the incentives of the ASes, currently implicit, explicit to each other and to end-users. Furthermore, they have transformed peering relationships, where appropriate, from implicit to explicit relationships. In Section 3.5, we show that the implicit nature of the relationships was the root problems in all three examples. Therefore, to the extent that these architectures can be realized, they are capable of solving this problem. Beyond this, it is possible to argue that we now have stability even in the repeated game. By re-introducing stability and resolving the conflicts, we create a framework in which ASes are willing to support user-directed routing.

Now we examine the costs at which these benefits have come. One potential downside is that we could have introduced significant complexity into the system through new or modified protocols. The use of routes as the good and the leveraging of protocols that already deal with routes allows us to suggest, in Section 3.6, that we did not do this. Another source of cost is that the ISPs must now track usage with finer granularity. However, this can be implemented solely at the ISP border, is becoming more supported in routers [49], and can be limited to those routes where the added monitoring is worthwhile. We note that an ISP must already today monitor the traffic of its peers, to ensure that it is making the right peering decisions. Finally,

we note because the business relationships are now in-band, the security, robustness, and auditability of the routing system becomes even more important. However, these points are already of great importance today. Nonetheless, we discuss some of these points in the next section.

An important point to note is that these changes and – and likely should – not precipitate an visible impact to normal individual users. This stems from the answer to the question "Who are the users?" above. While it is possible and reasonable to ask an overlay or an ISP to reason about routes and tradeoffs, this may be too complex a concept for individual users. Further, users are likely ill-informed of the different types of traffic their actions generate. Therefore, we envision that these incentives, while shared between neighboring networks and between networks and overlays or CDNs, will not in turn be exposed to typical end-users.

Finally, in response to the problems of Section 3.3, we argue that this scheme makes routing in a BGP-framework *simpler and more efficient*, even in the absence of user-directed routing. While ASes may (and likely will) continue to implement a lower or lowest-cost routing policy, the clarity of incentives will prevent perverse routing pathologies designed to maintain odd business relationships. Furthermore, the complexity of reducing cost through inter-domain routing can be significantly reduced, which in turn can decrease the cost structure of ISPs.

## 3.8 Discussion and Additional Concerns

Having seen the potential benefits and practicality of such a mechanism, we can examine several other key and open research questions. Some of these questions will be the foundation of the next chapter of this thesis. For the others, which we do not explore in more depth in this thesis, we offer our thoughts and preliminary analysis.

- *Who is the user?* In Section 3.6.2 we pointed out that there exist a range of potential hybrid mechanisms. In Section 3.6.1, we presented a spectrum of answers and we believe that a continuum of implementations is not only optimal but also presents a plausible adoption path. In particular, it is unclear that every end-user will want to be making these decisions. Thus, we suspect that the "end-user" in our models will primarily be the access ISPs and/or the overlays. These entities in turn can have relationships with end users where the tradeoffs are more manageable or well understood. Note that this issue is intimately related to the question of how to make the system scale – an

important question for any user-directed routing system.

- *Who absorbs the uncertainty?* Building out or maintaining capacity is not cheap, and there already exists a mini-tussle between players seeking longer-term contracts and players seeking more flexibility. User-directed routing brings this more into focus. Uncertainty over traffic volumes will exist in any system – but who should absorb the uncertainty? Should we look to 3rd parties such as CDNs? One compromise solution would be to employ user-directed routing on select paths for a fraction of one's traffic (e.g., the important flows). For even this simple approach to be accepted, the players must be properly informed and incentivized with a mechanism such as the one presented in this chapter.

- *What of privacy?* Today networks generally try to keep the details of their business relationships private. This is greatly reduced in an explicit incentive model. An interesting question is *"How much privacy is really lost in going from today's implicit model to an explicit model?"*.

  While future research in this area may be fruitful, it is important to note that the proposed architecture here does not necessarily pose a significant challenge to privacy. First, the overall structure of contracts today can be inferred in various manners. For example, Subramanian *et al* show how many business relationships can be inferred from public BGP feeds [96] and the proposed HLP architecture "explicitly publishes the provider-customer relationships" [97]. Furthermore, NDAs are known not to be perfect and much of the business structure is known to the large players in the Internet today. Second, information flow today, and in the proposed models, is strictly *bilateral.* Any firm, when pricing a good, faces the decision to expose underlying cost structures to the consumer (and potential downstream consumers) or to mask this information. Today's Internet is no exception: on-net vs off-net, international vs domestic pricing, as well as peering relationships are manifestations of this tradeoff. To the extent that it is significant, the relevant customers may in turn expose some of these differences to its customers. *In the architecture proposed in this chapter, this dynamic is completely unchanged.* While a naive implementation simply passes along (floods) all information, if an ISP is worried about the sensitivity of certain information, it can choose to simply not expose it.

- *Incentive Compatibility* We have simplified the logic for forwarding, but have not addressed price setting. In [32], Feigenbaum *et al* address this question and

present a strategyproof mechanism based on BGP. However, there are several strong assumptions in their model. Can we relax any of these while minimizing the strategizing of the players? We consider this question in the next chapter.

- *What are appropriate models?* Gao and Rexford's model, while quite simple, was very valuable to the analysis of this chapter. It provided both a framework and the important property of stability. Further, it provided a structured means of explicating the impact of user-directed routing.

  We however also saw that this model was not perfect, raising the question "*What types of models will be most useful in reasoning about incentive-based routing systems?*" There are a massive number of parameters to consider. We offer a few points. First, we feel strongly that instead of invoking the nebulous notion of (heterogeneous) policy, *money – a universal motivating factor – should play a central role.* Second, heterogeneity should be pushed out toward the edge, where users' preferences, particularly among applications, vary widely. Furthermore, we believe that there are gains to be made from bridging the gap between protocol specific models and idealized models of routing, such as the ones used in the Price of Anarchy analysis [86] [53].

We address these last two points further in Chapter 4.

## 3.9 Additional Related Work

Our research is not the first to suggest the notion of prices in routing. Indeed, several others have considered theoretical frameworks (e.g., Kelly *et al* [57]) for pricing and practical implementations (e.g., [70], based on BGP). Each of these works supports our motivation for considering prices. However, our analysis and proposal is different in several key ways. Unlike the work of Kelly, our proposal is designed to facilitate inter-network communication. Unlike [70] and other such proposals, our analysis and argument starts with the motivation of user-directed routing, presents an approach applicable to a number of routing architectures, and further does not argue for the introduction of incentives into BGP *per se.*

Alternative architectures and protocols have been proposed to address some of BGP's shortcomings in other manners. Some of these are discussed in the chapter. Others, such as the notion of introducing a Routing Control Plane (RCP) [31, 11], take an architectural approach to simplifying the logic in individual routers. RCP

is complimentary to the ideas of this chapter in that it provides a clean central location for the management of the incentive and pricing information. As such, it can significantly aid networks. A key difference in approach is that while the RCP and such approaches are designed to support a very rich policy space, the notion of incentives recognizes the fact that in most cases in practice the policy of ISPs is driven by the simpler strategy of profit maximization.

## 3.10 Summary

In this chapter we outlined the fundamental tussle between user-directed routing and ISPs, and demonstrated how this results from the current routing mechanism. User-directed routing has turned inter-domain routing into a meaningful repeated game which must be captured in our models and addressed in our protocols and architectures. To address this problem, we proposed the notion of incorporating prices into the routing system. We then demonstrated, through applications to various routing architectures, that this can be achieved with minimal technical steps and may instead simplify the system.

# Chapter 4

# Using Repeated Games to Design Incentive-Based Routing Systems

In this chapter we examine implementation parameters and details important for an incentive based routing protocol. Our investigation here is inspired by the previous chapter, which argues that the current economic policies of the Internet and the emerging technology of user-directed routing motivates the incorporation of prices into the routing system. The importance of the parameters analyzed is derived from the underlying economic factors governing the behavior of the autonomous players, in this case the competing networks.

We view the exchange of pricing information at an interconnect as a *repeated game* between the relevant players. Using such a model we are able to descry the impact that various protocol parameters – such as protocol period, minimum bid size, and unit of measure – have on the equilibrium outcome. Our analysis of these often surprising relationships enables protocol designers to appreciate and leverage these seemingly benign parameters, a result that has direct practical importance.

## 4.1 Introduction

As discussed in Chapter 3, Internet routing is a dramatic example of the introduction of economic concerns into an already rich design space. Traditional design concerns include the impact of system parameters on important objectives such as convergence, robustness, efficiency, and performance. Economic considerations now play a

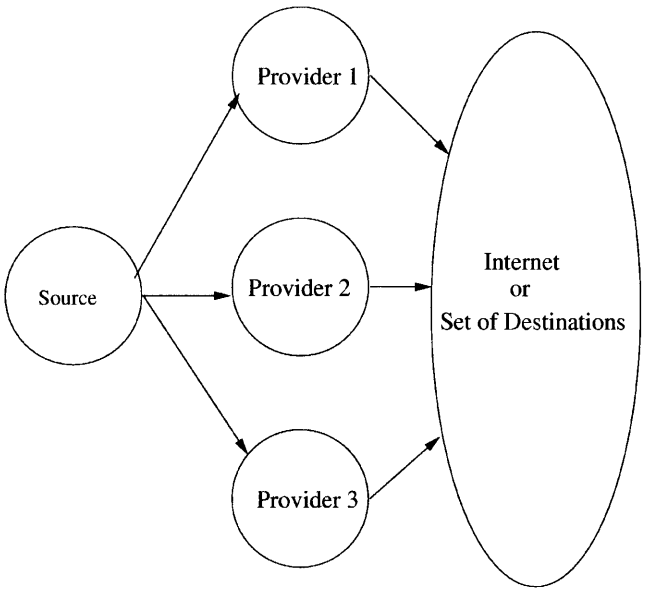



Figure 4-1: Three Networks Offering Connectivity for a Set of Routes to a Single Route Selector

chief role in the routing of traffic in today's Internet. Each Autonomous System (AS) is an independent profit-maximizing firm, competing to generate profit by routing bits on its network. Today this market plays out on two very different timeframes. On a multi-month or year timeframe, networks and customers negotiate economic contracts. Then, on a timescale of seconds, routers route traffic based on their configuration which encodes these business relationships.

The previous chapter presented several reasons to couple these two processes more tightly. These stemmed from problems with inter-domain routing today which are being exacerbated by user-directed routing. While the sample architectures proposed in Section 3.6.2 address the high level problem, there are several important practical concerns that must be addressed before realizing such an architecture.

In this chapter, we examine one of these technical questions: **"How should one design a protocol to convey pricing information for routes?"** We consider the interaction between a customer and a set of networks. The customer could be an enterprise, a Content Delivery Network (CDN), or an access ISP. The customer connects to multiple networks with each network providing connectivity to the same set of destinations, as in Fig. 4.1, and competing for the business.[1] Our examination of this question reveals that unavoidable yet seemingly minor design choices have significant practical effects. These parameters include the protocol period (the time

[1]For this reason, we use the terms 'network', 'AS', and 'player' interchangeably in this chapter.

between updates), the width (number of bits) of the price field, and the unit of measurement (e.g., megabits or megabytes).

Our examination of a particular interchange contrasts with some recent theoretical work on routing which considers general network topologies. There are several reasons why we feel this is the correct model. In practice, the customer does not pay every ISP in the route, but only the first provider. Therefore, the competition is in practice only between these providers. Such a model therefore maintains the bilateral economic nature of the Internet and is very similar to other applied work, including [43] and [92]. Further, to the extent that incentive-based routing is deployed, it will most likely be applied at interchanges where it is particularly useful (e.g., for CDNs who have a dramatic ability to shift traffic). They will not necessarily be used throughout the Internet graph. These points are discussed in more detail in the previous chapter.

The analytical framework we apply to this problem is that of repeated games. As discussed in Chapter 2, the the threat (or promise) of future behavior can impact current actions, and therefore the equilibrium of one-shot and repeated games can differ significantly. This makes repeated game analysis a useful and important analytical tool for a distributed protocol that will be implemented by autonomous entities. In this particular example, the importance of the repeated game is that the threat (or promise) of future behavior can impact current actions. In our example, knowledge that a competitor will react (e.g., by matching price) in the future impacts the way a network sets price in the current period. By examining the repeated context, our work is in stark contrast to most prior work in this space, namely the celebrated Feigenbaum, Papadimitriou, Sami, and Shenker (FPSS) [32] analysis.

In our model of inter-domain routing, the dynamics of repeated games cause certain protocol parameters to achieve significant importance. It is well-known that in the repeated game, there exist parameters that significantly impact the equilibrium outcome. However, routing is special due both to the particulars of the problem and that it transpires via a fixed network protocol. Therefore, the contributions of this chapter include not only a repeated model for routing but also formal analysis of the domain-specific parameters relevant to routing. We summarize those results in the form of practical statements:

1. A longer protocol period (a slower protocol) can lead to a lower price.

2. Using a more granular format (e.g., megabits instead of megabytes) can lead to a higher price.

3. A wider price field in the protocol can lead to a lower price.

These relationships are significant. Given this sensitivity, we also show how protocol designers, to the degree desired, can bound prices and their sensitivity to repeated game effects. **These conclusions have clear, direct, and previously unrecognized practical significance for protocol designers.** Further, these relationships are meaningful in that they can, to the extent desired, constrain the prices and the potential outcomes. As such they solve, to some degree, the problem presented by repeated games – namely the large number of equilibria and potential outcomes.

The rest of this chapter proceeds as follows. In Section 4.2 we examine the impact that repeated games have on routing, and why this may be a problem. In Section 4.3 we present a model of repeated routing. We then analyze this model for a particular equilibrium strategy, price matching, deriving the above conclusions. In Section 4.5.1 we then generalize these results to a larger class of strategies whose only constraint is that the punishment be at most proportional to the deviation. We then continue, in the rest of Section 4.5, to consider additional relaxations and generalizations to our model, such as asynchronous play, confluent flows and multiple destinations, before ending with a discussion of the results.

## 4.2 The Challenge of Repeated Routing

The inherently repeated nature of inter-domain routing plays a significant role in the outcome of the system. Consider a simplified routing game, as depicted in Fig. 4.1. This game presents the same phenomenon as the repeated Prisoners' Dilemma example presented in Chapter 2. Under reasonable assumptions, firms can maintain an artificially higher price if their strategies include appropriately crafted threats to punish deviators.

It is important to contrast this repeated context with prior work on routing, namely the celebrated work of Feigenbaum, Papadimitriou, Sami, and Shenker (FPSS) [32]. They demonstrate that it is possible to implement the well-known Vickrey-Clarke-Groves (VCG) mechanism efficiently with a protocol that resembles BGP.[2]

Our work builds upon their results by considering their model in the repeated game. Thus we summarize it here:

- A set of nodes N, with $n = |N|$, representing the ASes
- A constant per packet cost $c_i$ for each node i

[2]FPSS were not the first to consider the VCG mechanism for routing [74, 45]. However, one key contribution of the FPSS work is the framing of the problem with the nodes as strategic agents. This maps to the problem of AS competition and motivates us to consider the repeated game.

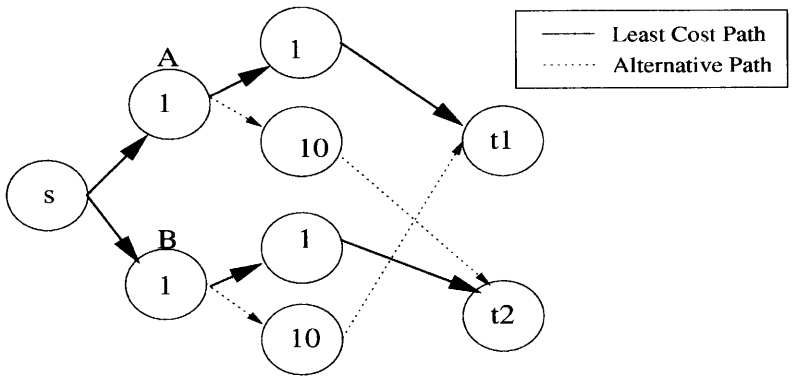



Figure 4-2: A Topology where the Repeated FPSS Model is Not Strategyproof in the Repeated Game

- A traffic matrix $T_{i,j}$ which is exogenous and fixed (i.e., inelastic demand)
- Each AS has infinite capacity

In this context, the FPSS mechanism ensures that the Least Cost Path (LCP) is selected by incenting each network, although still selfishly motivated, to bid a price equal to its true cost. Therefore, they realize the social choice function and create an environment where the networks have no incentive to strategize. More formally, truth telling is weakly dominant, implying that the mechanism is strategyproof. Here a strategyproof mechanism is desirable since it means that networks do not need to spend time and effort strategizing about prices. Furthermore, the FPSS mechanism is based on BGP, suggesting (along with scalability analysis presented in [32]) that the mechanism could be implemented in practice.

The VCG mechanism, and thus the FPSS implementation, obtains its strategyproof property through a carefully selected payment to each node. Each node, $i$, on a Least Cost Path (LCP) between a source-sink pair $(s,t)$ is paid $c_i$ plus the difference between the cost of the LCP and the cost of the LCP if $i$ did not exist. For example, in Fig. 4.2, node A is on the LCP from s to $t_1$. For traffic from $s$ to $t_1$ A is paid:

$$p_A = LCP_{(c_A=\infty)} - LCP + c_A = (10+1) - (1+1) + 1 = 10$$

Similarly, B is paid 10 for each traffic unit from s to $t_2$.

However, it is well known that VCG mechanism is not strategyproof in the repeated game. If A and B both bid 20 until the other defects, each will be paid:

$$(20+10) - (20+1) + 20 = 29$$

We can easily show that it is possible for this to be an equilibrium for sufficiently patient players. More formally, there exists a $\bar{\delta}$ such that for all $\delta \geq \bar{\delta}$ this strategy can exist in equilibrium.[3] This demonstrates that although Internet routing is a repeated setting, the VCG mechanism (and thus the FPSS implementation), is not strategyproof in the repeated routing game.

Said differently, it is known that the VCG mechanism is susceptible to collusion. But in the one shot game, without explicit outside agreements, such cooperation is not possible. In the repeated game, the subsequent periods provide the players with a means of obtaining a higher price without any explicit collusion, side-payments, or constructs of any sort.[4] This is very troubling since routing is clearly a repeated game, not a one-shot game.

## 4.3 A Model of Repeated Routing

The observation the mechanism is not strategyproof in the repeated game is is worrisome for several reasons. First, to the extent that the VCG/FPSS prices are fair or desirable, we have no way of ensuring that they will occur. Second, we do not initially have any understanding of what the outcome will now be. Third, we do not understand how design decisions will impact the outcome.

To address these questions, we analyze a model of the repeated game. First, we present the formal model. We then analyze the model to determine the equilibrium outcome. Finally, we analyze this outcome to descry the impact of the fundamental design parameters on this outcome. We do not seek to impose a particular outcome (e.g., minimize price) on the system, since as we discuss in Section 4.6, it is unclear there is a universally correct and acceptable answer. Instead we focus on understanding these unavoidable parameters.

### 4.3.1 Key Intuition and Analytical Approach

Before delving into the model, analytical framework, and mathematics, we first present the high level intuition which underlies the analysis presented in the remainder of the chapter. Consider a small number of homogeneous firms competing for an amount of traffic. At any point in time, each firm faces a key strategic decision. One

[3]For completeness, in the one-shot game, bidding 20 is not an equilibrium strategy. The other player can bid 11 and get all the traffic on both routes for a price of 29 – yielding a higher profit.

[4]Certainly, with additional such collusive constructs, other equilibria are possible. We ignore those for the purposes of this chapter.

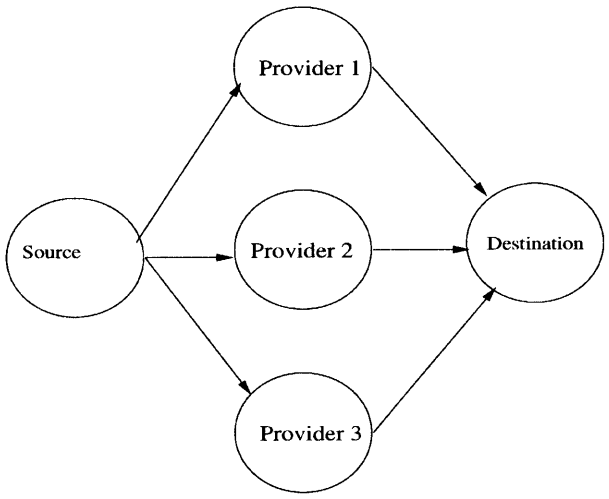



Figure 4-3: A Depiction of the Repeated Incentive Routing Game

strategy is to be the low cost provider and receive all the traffic. Another is to offer a higher price to the market, somehow splitting the traffic with the other providers – but potentially garnering more profits due to the increased price.

The firm's willingness to take this second, cooperative, strategy is a function of several factors. One factor is the *granularity of the action space* – which in this example is the amount by which the firm needs to deviate to get all of the traffic. For example, if the current market price is \$100 and the firm can get all the traffic with a bid of \$99.99, it will be more likely to deviate than if somehow it were constrained to offering integer prices (e.g., \$99). Another factor is the *discount factor* which here manifests itself in several ways, notably in the length of the game. If the firm feels that the game will end soon, it will be more likely to decrease price to get the extra profit. On the other hand, if it feels the game will last a longer period of time, it may not want to perturb its competitors.

A key insight of this chapter is that in practice these factors are directly determined by parameters of the protocol. In particular, the width of the pricing field and the representation used determines the granularity of the action space. Further, the protocol period determines the number of periods the game will be played and thus the effective discount factor.

Because this intuition is fundamental to the competitive dynamics of the situation, the results obtained are robust to a wide-range of practical and important assumptions. These include, but are not limited to:

- Heterogeneous networks

- Asynchronous protocols
- Multiple destinations in a network
- Multi-hop networks
- Confluent (BGP-like) routing
- A wide class of rational strategies for the firms

In the interest of clarity, however, we do not present a general model that contains all of these properties. Instead, we first start with a simple model that captures the essence of the game, provides for lucid analysis, and demonstrates the key intuition. After, in Section 4.5, we return to these assumptions and formally prove the same set of results for models that incorporate these more sophisticated assumptions. We also discuss some potential future extensions in Section 4.6.

### 4.3.2 The Repeated Incentive Routing Game (RIRG)

Our model is based on the FPSS model, presented above. To their model we make extensions to capture the repeated nature of the game and the properties of the protocol. We also introduce some simplifying restrictions to make the model more tractable for the initial analysis. In Section 4.5 we relax and address many of these assumptions.

The game analyzed in this section is depicted in Fig. 4-3.

**Repeated Incentive Routing Game Model**

- There is one source and one destination.
- Each of the $N$ networks connects directly to both the source and the destination with exactly one link, as depicted in Fig. 4-3.
- Each network has an identical, constant per-packet cost $c$ for transiting the network[5], identical quality, and infinite capacity.
- Bids are represented as fields in packets and thus are discrete. The maximum granularity of the representation, equivalent to the minimum change in a bid, is represented by $b$. For simplicity, we assume that $c$ is a multiple of $b$.

[5]We note that this maps cleanly to average-based billing, a common billing technique in practice. A richer discussion of volume-based versus rate-based models and their prevalence in industry is beyond the scope of this chapter.

- Each AS is perfectly-patient with respect to the time value of money.
- The low cost bid in each period is common knowledge. Specifically, before the next time an AS advertises a price, it knows the lowest price bid in the prior round.
- The game is finite. The game length is represented as an exponential random variable with mean $D$. $E(D)$ is known to all players but the actual value is unknown to the source or any of the other ASes. The duration corresponds to the expected period of time for which the other factors will be stable.[6]

**Play of the Game**

1. The game proceeds in a series of rounds, each of length $d$, a constant of common knowledge. For simplicity, we assume that $D$ is a multiple of $d$. Thus, we can relate $d$ and $D$ as:

$$d = D(1 - \delta) \tag{4.1}$$

   where $(1 - \delta)$ is a constant representing the per-period probability of the game coming to an end.

2. At the start of each round, each of the $N$ players advertises its per-packet price simultaneously.
3. For the entire period, traffic is routed over the provider with the lowest price. In the event of a tie, traffic is split evenly among the providers with the lowest price.
4. Each provider is paid for the number of packets that transit its network. The rate paid for each packet is the price it advertised at the beginning of the round (first price auction).

### 4.3.3 Equilibrium Notion and Strategy Space

Since the set of potential strategies and equilibria for this game is quite large, it is important to refine the space to a set of strategies only containing those which are potentially reasonable. Our first significant refinement of the equilibrium notion is to consider only strategies which are subgame perfect. Subgame perfection, defined

[6]The property of having stability for a sufficiently-long, finite, and unknown period of time corresponds very well to the true nature of Internet interconnects.

below, means that in every stage game, all players must play a strategy that is optimal, given the remainder of the game. It thus takes the entire discounted stream of payments into account, precluding myopic strategies but permitting long-term thinking. This is a very reasonable restriction and one standard for repeated games. Also, for clarity, we restrict ourselves to symmetric equilibria, where all players play the same strategy, and pure strategies. This allows us to speak of a single strategy being played.

**Definition 1.** ***Subgame Perfection:*** *A strategy $\alpha$ is subgame perfect if i) $\alpha$ is a Nash equilibrium for the entire game and ii) $\alpha$ is a Nash equilibrium for each subgame.*

In a repeated, simultaneous-move game such as ours, the set of Subgame Perfect Equilibria (SPEs) can still be quite large. One class of strategies are "trigger price strategies" [44]. In this context, players offer some desirable price, $p^*$, so long as all other players do. If a player deviates, offering some $p' < p^*$, the other players *punish* the deviating player by playing some $\hat{p} < p^*$. In general, trigger price strategies allow for the players to return to $p^*$ after some period of time. The intuition of these strategies is that in equilibrium the threat of punishment can maintain a higher price.

While there is empirical studies supporting the existence of such behavior, such a severe and coordinated practice may seem implausible in many contexts. For example, in the bandwidth market, we have not observed such wild swings. Instead, as costs decrease and competitive pressure has increased, we have seen prices move down rather smoothly and steadily. Such a phenomenon may be better modeled by a *price matching* strategy where players play the lowest price observed in the prior period.

The key difference between these two classes of strategies is how we perceive the reaction to a deviation. To the extent that it is a *punishment*, trigger-price strategies are appropriate. To the extent that it is simply a *protective reaction or learning mechanism*, price matching seems more appropriate. Price matching may even be too severe, as more appropriate strategy may be to price match for a certain period before returning to $p^*$. However, in all strategies, deviation of a player leads to decreased profit for some number of future periods.

We can generalize this space of strategies. In particular, in a parameterized space, price-matching is a mild punishment for an infinite amount of time. By parameterizing the punishment time and severity, we can consider a larger class of strategies. As discussed, for the sake of clarity we initially discuss and analyze price matching strategies. However, in Section 4.5.1, we show that our results hold for a much larger class of strategies, namely all strategies where the punishment is no greater than a

constant multiple of the deviation. This definition likely encompasses any strategy that an ISP would find reasonable. It not only permits more severe punishments but also permits strategies that forgive and return to $p^*$.

We are now ready to formally define the price-matching strategy:

**Price Matching (PM) Strategy**

S1) At t = 0, offer some price $p_i^0$ For $t > 0$:

S2) $p_i = max(c, \min_j(p_j^{t-1}))$

where $p_j^t$ is the price offered by player $j$ in period $t$ such that PM is SPE.

To construct PM as a SPE, we will use the One Stage Deviation Principle which states that a strategy is a SPE if and only if it is not possible to profitably deviate in exactly one stage-game. This allows us to consider simple one-stage deviations as opposed to more complicated multi-stage deviations. We state the principle, whose proof can be found in [40], below:

**Theorem 1 (One Stage Deviation Principle).** *In an infinite horizon multi-stage game with observed actions where the payoffs are a discounted sum of per-period payoffs and the per-period payoffs are uniformly bounded; strategy profile $\alpha$ is subgame perfect if and only if it satisfies the condition that no player $i$ can gain by deviating from $\alpha$ in a single stage and conforming to $\alpha$ thereafter.*

We can now use the intuition from this theorem to construct PM as SPE. (We will formally invoke the principle in Theorem 4.3.3 below.) While the second step of the strategy is clear, it is not immediately obvious how a player should select the initial $p_i^0$. From the principle, we have that the player cannot benefit from deviating. From price matching, we have that in equilibrium $p^t = p^0 \ \forall t$. (Thus, we drop the superscript notation and simply write $p$.) This means that if price matching is an SPE then:

$$\sum_{t=0}^{\infty} \delta^t \pi_i(p, p) \geq \pi_i(p - b, p) + \sum_{t=1}^{\infty} \delta^t \pi_i(p - b, p - b) \tag{4.2}$$

for a given $(\delta, b)$. Just as Eqn (2.1) simplifies to (2.3), we can simplify Eqn (4.2) to:

$$\pi_i(p, p) \geq (1 - \delta)\pi_i(p - b, p) + \delta \pi_i(p - b, p - b) \tag{4.3}$$

Informally, this condition says that we will accept p only if the payoff to playing p forever is greater than the payoff from deviating once and suffering the consequences. The $\pi_i(p, p)$ term represents the payoffs of playing $p$, $(1-\delta)\pi_i(p-b, p)$ is the weighted

payoff to deviating by some amount b, and $\delta\pi_i(p-b, p-b)$ captures the payoffs in the future. Note that it is strictly dominant to make the smallest possible deviation from $p$. Thus, we use $b$, the size of the minimum bid change, as the magnitude of the deviation without loss of generality.

Of all the values of $p$ that satisfy Eqn (4.3), we consider the profit-maximizing value, which we define to be $p^*$. Therefore,

$$p^* = \max_p \text{ s.t. } \pi_i(p,p) \geq (1-\delta)\pi_i(p-b,p) + \delta\pi_i(p-b,p-b) \tag{4.4}$$

(We solve for $p^*$ explicitly for our game by expanding $\pi(.)$ in the following section.)

We now seek to show formally that Price Match is sub-game perfect.[7] This relies on the One Stage Deviation Principle and our construction above.

**Lemma 1.** *PM is a SPE.*

*Proof.* First we note that the RIRG game satisfies the technical conditions of the principle and the fact that the game is finite ensures that the discount factor, $\delta < 1$. Therefore, we can apply the theorem and consider only one-stage deviations. We look at each stage of the specified strategy:

S1) By construction, assuming that other players offer $p^*$, it is optimal to offer $p^*$. By definition of $p^*$, bidding a lower value decreases the discounted stream of profits. A higher price leads to no profits in this period and no prospect of higher profits in the future.

S2) Again by construction, there is no benefit to decreasing price. Likewise, increasing price given that others are playing PM does not help.

Since we have examined all one-stage deviations, we have that PM is a SPE. □

### 4.3.4 Analysis of the Model

With a model and equilibrium notion, we can now examine the equilibrium conditions. The first step is to derive an explicit expression for $p^*$ in terms of the parameters of the game. [8]

[7] It is important to note neither that the definition of SPE nor this proof provides any guarantee that the value to which the players converge will indeed always be $p^*$. One benefit of price matching is that it seems likely that players will converge in a continuous fashion to $p^*$.

[8] As discussed, price is discrete. However, for notational simplicity, we analyze the continuous variable $p$ such that the market price is $\lfloor \frac{p}{b} \rfloor b$.

Table 4.1: Summary of Key Terms

| Term | Meaning |
|---|---|
| N | Number of firms competing for the traffic |
| b | Minimum bid change size |
| p | Price |
| $p^*$ | Profit maximizing price |
| $\pi(.)$ | Per-firm profit function |
| $\delta$ | Per-period chance of the game ending |
| $d$ | Period of the protocol |
| $D$ | Expected stability of network topology |
| $T$ | Total amount of traffic |

**Theorem 2.** *In the RIRG, the unique equilibrium price when all players play Price Matching is given by:*

$$p^* = \frac{b\left(\delta bN - \delta b - N\right)}{1 - N + \delta N - \delta} + c$$

*Proof.* Since the firms seek to maximize their profits, we consider the profit-maximizing price matching strategy, which bids $p^*$ as given by Eqn (4.4). Restating, we have the market price, $p$, satisfies:

$$\pi_i(p, p) = (1 - \delta)\pi_i(p - b, p) + \delta\pi_i(p - b, p - b) \tag{4.5}$$

where p is the price advertised.

Define $m = p - c$ for notational simplicity. We now expand $\pi_i$ based on the definition of the game:

$$\pi_i(p_i, p_{-i}) = \begin{cases} \left(\frac{T}{N}\right) m, & p_i = p_j, \forall i \neq j \\ T * m, & p_i < p_j \forall j \neq i \\ 0, & otherwise \end{cases}$$

This yields:

$$\left(\frac{T}{N}\right) m = (1 - \delta)\left(m - b\right) T + \delta \left(\frac{T}{N}\right)(m - b) \tag{4.6}$$

Solving, we have:

$$m = \frac{b\left(\delta N - \delta - N\right)}{1 - N + \delta N - \delta} \tag{4.7}$$

or

$$p = \frac{b\,(\delta N - \delta - N)}{1 - N + \delta N - \delta} + c \tag{4.8}$$

Since all players are homogeneous and since we consider only symmetric equilibria, this is thus the unique equilibrium. □

## 4.4 Understanding the Result

Given an expression for the equilibrium price, we turn to the practical questions that we seek to understand.

### 4.4.1 Protocol Period

We examine the model parameter tied to the period, $\delta$, holding the other factors (including D) constant. Intuitively, it may seem that the period of the game should have no impact on prices. Alternatively, a shorter period–corresponding to a faster protocol–would perhaps help to keep the market more competitive. This is not necessarily the case.

**Theorem 3.** *The protocol period and the market price are positively correlated – or* $\frac{\partial p}{\partial \delta} > 0$.

*Proof deferred to appendix.*

Recall now that $d = D(1 - \delta) \rightarrow \frac{\partial d}{\partial \delta} < 0$. This coupled with Theorem 3 yields

$$\frac{\partial p}{\partial d} < 0 \tag{4.9}$$

In other words, *as the protocol period increases, the price decreases – a surprising and initially counterintuitive result!*

Careful consideration provides us with the rationale behind this conclusion. When a player deviates, it enjoys a one-time increased payoff at the expense of diminishing the future stream of payoffs. Consequently, the longer the period is before a competitor can match the price, the bigger the benefit to deviating. Furthermore, a longer period means fewer expected future periods. As a result, as we increase the protocol period, we increase the propensity for a player to lower its price. It is well-known in the repeated game theory of oligopolies that fewer periods can increase price. But it is interesting to realize that the protocol period, typically analyzed in the context of

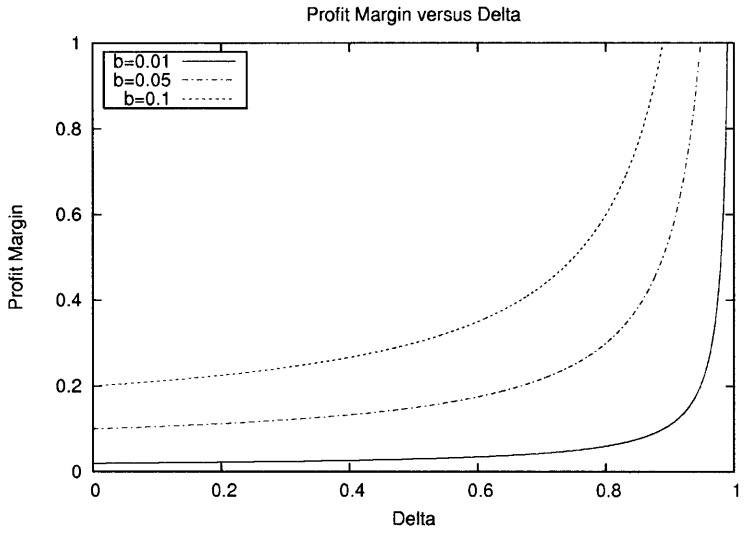



Figure 4-4: Price as a function of $\delta$ for $N = 2$, $c = 0$, and $b \in \{0.01, 0.05, 0.1\}$. Margin increases with $\delta$ and and is very sensitive to $\delta$ when $\delta$ is large.

information flow and convergence, in practice also defines the number of rounds and thus significantly impacts the equilibrium.

We depict this relationship between $p$ and $\delta$ by graphing Eqn (4.8) in Fig. 4-4. As can be observed, $p$ is strictly increasing in $\delta$ but converges readily to $c + \frac{Nb}{N-1}$ as $\delta \to 0$. [9]

Although this phenomenon may seems counter-intuitive at first, most consumers are familiar with it in the form of "Price Match Guarantees" offered by many major retailers [95] [80]. While the policies vary, the notion is that Firm A will match any competitor's advertised price that is lower than A's price. While there are other factors at play in these markets, this practice can be abstracted in the notion of a protocol period. Instead of waiting some period to match the competitor's price (e.g., in the next week's circular, in the following day, etc.) a price match guarantee effectively brings the period to zero. Once a firm lowers its price, the other firm effectively matches price immediately. Thus, one result of these policies is to dissuade competitors from lowering price, since, it can be argued, it will not provide that competitor with any additional revenue.

[9] For $N > 2$, $c + \frac{Nb}{N-1}$ yields $c + b$ when discretized to a multiple of $b$. In the one-shot game bidding $c$ or $c + b$ are both Nash Equilibria. We return to this subject in Section 4.5.3.

Table 4.2: Impact of Protocol Parameters on Price

| Variable | Impact on Price |
|---|---|
| $N$ : Number of players | Decreases |
| $b$ : Minimum bid size | Increases |
| $d$ : Period of the protocol | Decreases |
| $D$ : Stability period for the topology | Increases |

Table 4.3: The Impact of a $1 Price Change with Megabyte and Megabit representation formats

| Format | Traffic | Price | Revenue | New Price | New Revenue |
|---|---|---|---|---|---|
| Mbits | 1000 | $100 | $100,000 | $99 | $99,000 |
| MBytes | 125 | $800 | $100,000 | $799 | $99,875 |

### 4.4.2 Additional Parameters

We now consider the the other relevant parameters in similar fashion. For each parameter found in the expression for the equilibrium price, we present the main result and some intuition to provide better understanding. The results are summarized in Table 4.2.

#### Minimum Bid Size, b

Similar to the analysis of period, we can show that $\frac{\partial p}{\partial b} > 0$ – as we increase the minimum bid size, the equilibrium price increases. This again comes from the firm's decision which weighs the one-time benefit of deviating versus the longer-term cost. The less a firm is able to decrease $p$ and still get all the market, the more profit it garners in the short-term and the less punishment it suffers in the long term. Therefore, it is more likely to deviate.

It is important to understand that this is more than just a matter of precision. Fig. 4-4 plots equilibrium price versus $\delta$ for $N = 2$, $b \in \{0.01, 0.05, 0.1\}$. Note not only that $p$ changes significantly but moreover that the change in $p$ is qualitatively greater than the change in $b$.

Generally in practice, the minimum bid size is not an explicit parameter but rather it implicitly manifests itself in two protocol parameters. The first is the width of the pricing field. In most any protocol this width is likely to be fixed. Here we see that *increasing the width of the pricing field can decrease the price in the system.* Another

means by which the minimum bid size manifests itself is via the unit of measure. Given a fixed granularity on prices, it makes a difference if we represent quantities in megabits or megabytes. For example, consider a system in which prices are set at whole dollar increments. Using mega*bytes* as opposed to mega*bits* provides for larger price values and thus more granularity in the prices, holding all other parameters constant. In Table 4.4.2 we see that a \$1 decrease when using megabits causes a 1% decrease in revenue whereas a \$1 decrease when using megabytes causes a 0.125% decrease in revenue. Per the logic outlined above, we see that *using megabytes instead of megabits can lead to a lower price.*

### Stability Period, D

The stability period is likely not under the control of the protocol per se, but it is still useful to understand its impact on prices, holding other parameters (including d) constant. We have that $\frac{\partial p}{\partial \delta} > 0$. Since $d = D(1-\delta) \rightarrow (1 - \frac{d}{D}) = \delta$ we have that $\frac{\partial \delta}{\partial D} > 0$. Thus, $\frac{\partial p}{\partial D} > 0$. This should come as no surprise given the prior two examples. As we increase the expected duration of the game, the relative importance of the stream of future payoffs increases. Thus, a player is less willing to deviate.

### Number of Players, N

While the number of players is generally assumed to be constant, it is useful to note that similar to the other variables, we can show that $\frac{\partial p}{\partial N} < 0$. This conclusion is perhaps the most likely to be obvious *a priori.* As the number of firms increases, the profit is split among more players. Thus, as the number of firms increases, so too does the benefit from a one-stage deviation–and thus the propensity to deviate. This corresponds with the basic intuition that with more firms we approach perfect competition.

## 4.4.3 Discussion

Because the rest of the chapter consists of various relaxations and further analysis of the results presented above, we pause here to make a few observations:

- *There are several protocol parameters which – unexpectedly – may significantly impact the equilibrium price.* These include the protocol period, the width of the pricing field and the unit of measure. Unlike some properties that one might readily be able to identify and reason about (such as the number of players); *a*

*priori* it is unclear that these parameters have any affect. Further, it is unclear which way they push the equilibrium price. These often counter-intuitive results are therefore quite revealing.

- *The conclusions about these parameters are directly applicable to system design.* Understanding the impact of these parameters is a useful result. What is perhaps most important, however, is that *a priori* a protocol designer may not have even considered these parameters as relevant at all! Therefore, merely understanding that they are relevant, let alone understanding how they impact the equilibrium, is an important conclusion.

- *We have **not** shown that these repeated outcomes will always occur, but still believe consideration of the the parameters is important.* We have shown that it is possible to obtain increased prices in repeated equilibria, but have not shown that this result is robust to all variations to our model. In Section 4.5, we will consider various relaxations to our model and show that similar results can be obtained. However, in a general setting, it is possible to construct degenerate scenarios in which such increased prices are not possible. Therefore, our argument is not that increased prices will always result if the parameters are not considered. Instead, we argue that in general (and in changing) environments, such outcomes *may* result, and in some cases will *likely* result. In practice, it will be the rare case when one is certain that such outcomes will not occur. Since a good protocol should be applicable to a wide range of circumstances, we therefore believe that the protocol designer should and must take these parameters into account.

## 4.5 Extensions to the Model

In this section we show that the key intuition and the spirit of the results derived from the simple Repeated Incentive Routing Game (RIRG) hold in more general networks, strategies, mechanisms, or assumptions. There are many aspects of the model that are either simplified (e.g., one destination) or not always general (e.g., 1st price auction). However, the intuition from the simpler example carries over to far more general models. In this section, we present and discuss several of these modifications to the game. In all cases the core results – the impact of the granularity of the action space and its manifestation in the protocol parameters – holds. The underlying reason for this is that the key intuition outlined in Section 4.3.1 holds in all of the games.

Firms face a key decision: attempt to be the lowest-priced provider and take the whole market, or split the market at a higher price with multiple firms. For some extensions, where we feel that the proof is particularly insightful, we present a full formal proof of our results. In other cases, where the extension is relatively simple, we present only a discussion.

### 4.5.1 Generalizing the Strategy Space

There are several reasons why the strategy consider thus far, price-matching, may be too narrow a space. Perhaps most important, it assumes that prices never return to the original $p^*$. Further, the punishment phase is limited at $p - b$. For example, we could relax both of these assumptions and consider a set of strategies which punish at $p - kb$, $k > 0$ for $T$ periods before returning to $p^*$.

In this subsection, we generalize our results to a larger set of strategies, which we call *Porportional Punishment strategies.* The punishment of these strategies is porportional to the deviation, and we permit prices to return to some higher price, including $p^*$. This set of strategies is quite large and likely contains any strategy that an ISP would find reasonable. While it may also contain strategies that are not reasonable, we simply use it to show that it is a sufficent condition for our results to hold in general. The reason that we bound the punishment is to prevent the grim strategy (or variants thereof) where the punihsments can become arbitrarily large. With sufficently high $\delta$, this would enable infinitely high prices – in the absence of other constraints (such as an non-zero elasticity of demand) – which is not sensible.

For this set of strategies, we seek to understand the highest possible market price, $\bar{p}$, given a fixed $(N, \delta, b)$ tuple. We derive a bound and show that this bound is tight. Using this bound we can show that our conclusions regarding the parameters still hold. Further, we can show that it is possible for the protocol designer, if she indeed desired, to bound prices using the parameters to $p^I + \epsilon$ where $p^I$ is the price in the one-shot, first-price auction.

To begin our analysis, we introduce some new notation: Let $\alpha_h^t$ be the price proscribed by strategy $\alpha$ in period $t$ given history $h$. Let $p_i^t$ be the bid of player $i$ in period $t$. Finally, let $D(\sigma, h, s) = \{t > s \mid \exists\ p_i^t < \alpha_{h^{t-1}}^t\}$. These are the periods where there has been a deviation.

We now define $\Delta$ which represents the set of all one-stage deviations for a strategy $\sigma$.

**Defintion:** Let $\Delta$ be the set of all tuples $(h, \hat{h}, \sigma)$ such that:

1. $\sigma$ is a SPE strategy.

2. $h^t = \hat{h}^t \; \forall \; t < t_0$

3. In $t_0$, $\exists \; p' = p_i^t < \alpha_{h^{t-1}}^t$

4. $D(\sigma, \hat{h}, t_0) = D(\sigma, \hat{h}^t, t_0)$

Informally, this says that the two histories are the same until $t_0$ (2), there is a deviation at $t_0$ (3), and that there are no further deviations (4). We now define the set of *Porportional Punishment Strategies* (PP) as follows:
**Definition:** $\sigma \in PP_k$ iff $\sigma$ is a symmetric SPE and $\forall (h, \hat{h}, \sigma) \in \Delta$, $\forall t > t_0$, $\forall p'$:

$$\sigma_h^t - \sigma_{\hat{h}}^t \leq k(\sigma_{\hat{h}}^{t_0} - p')$$

This captures our definition of porportional punishments. We now turn to analyzing the equilibrium conditions. To do so, we introduce two new terms: $p^\alpha$ and $\bar{p_k}$:

**Definition:** $p^\sigma$ is the highest price obtained by $\sigma$ in any period, that is:

$$p^\sigma = \max_{\forall t,h} \sigma_h^t$$

**Definition:** We define $\bar{p_k}$ to be the maximum price for all strategies in $PP_k$. More formally,

$$\bar{p_k} = \max_{\alpha \in \mathrm{PP}_k} p^\alpha$$

**Theorem 4.** *If $\sigma \in PP_k$, then $p^\sigma \leq \frac{b(\delta N - N - \delta k)}{(\delta - 1)(N-1)}$. Further this bound is tight, that is, $\exists \; \bar{\alpha}$ such that $p^{\bar{\alpha}} = \bar{p} = \frac{b(\delta N - N - \delta k)}{(\delta - 1)(N-1)} + c$.*

*Proof deferred to Appendix*

Given this bound of $\bar{p_k}$, we can also analyze the bound just as we did for the bound we derived for price matching. Simply by taking the partial derivatives, we see that we get the same qualitative results and resulting intuition.

**Theorem 5.** *In the RIRG, if players play strategies in PP, the value of $\bar{p}$ varies with the parameters $(\delta, b, N)$ in the same manner as the optimal price matching price. That is: $\frac{\partial \bar{p}}{\partial \delta} < 0$, $\frac{\partial \bar{p}}{\partial b} > 0$ , $\frac{\partial \bar{p}}{\partial D} > 0$ , and $\frac{\partial \bar{p}}{\partial N} < 0$.*

Note that Theorem 4 corresponds to Theorem 2 and Theorem 5 corresponds to Theorem 3. Less formally, this means that the practical insight regarding the protocol parameters still holds even in this more general context where we know much less about the behavior of the networks.

Finally, we can consider the implementation design question of how to limit prices.[10] Here we obtain a result that in practice, we can constrain the price inflation that may occur in the repeated game via use of the protocol parameters. More formally:

**Theorem 6.** *For an instance of the RIRG with N players playing strategies in PP, $\forall\ \epsilon > 0$ there exists a tuple $(\delta, b)$ such that $p_m < c + \epsilon$ where $p_m$ is the price realized in the market.*

### 4.5.2 Asynchronous Play

One place where RIRG model does not match the reality of Internet routing is the assumption of synchronized play. In real routing systems, not only is the message exchange not synchronized, but moreover synchronization would be a hard property to achieve, even if it were desirable. While synchronous play is the normal model for repeated games, a limited amount of recent work has explored asynchronous models of repeated games, albeit in other contexts [61, 102]. While the analysis of the previous section relied on this synchronized assumption, the key intuition of the problem (presented in Section 4.3.1) does not. Therefore, we are able to obtain essentially the same results in the asynchronous case, which we present below.

While the analysis below suggests that the asynchronized play does not change the game, that is not correct. Indeed, the asynchronized play has a significant impact on the set equilibrium strategies that can be played. For example, the grim strategy of setting price equal to cost in response to a defection is no longer a SPE strategy. We discuss this more in Section 4.6.2.

**Theorem 7.** *If $\sigma \in PP_k$, then $p^\sigma \leq p^\sigma \leq \frac{kb(-N\phi+N\phi\delta-\delta^N)}{1-N\phi+N\phi\delta-\delta^N}$ in the asynchronous game. Further this bound is tight, that is, $\exists\ \bar{\alpha}$ such that $p^{\bar{\alpha}} = \bar{p} = p^\sigma \leq \frac{kb(-N\phi+N\phi\delta-\delta^N)}{1-N\phi+N\phi\delta-\delta^N}$*

*Proof.* We know that in equilibrium:

$$\sum_{t=t_0}^{\infty} \delta^t p^\sigma \geq (p^\sigma - b)N + \sum_{t=t_0+1}^{\infty} \delta^t \beta_{(i,t)}(p^\sigma - b, t_0, \sigma_{-i})$$

[10] Of course, as we discuss in Section 4.6, limiting prices is not necessarily a desirable goal.

where $\beta(.)$ specifies the continuation payoff to player $i$ for a deviation at $t_0$ with other players playing $\sigma_{-i}$. Since $\sigma \in PP_k$, let us consider the most severe punishment (without loss of generality). This yields:

$$\sum_{t=t_0}^{\infty} \delta^t p^\sigma \geq \sum_{i=t_0}^{N-1} \frac{(p^\sigma - kb)N}{(i+1)} \delta^i + \delta^N \sum_{t=t_0+N}^{\infty} \delta^i (p^\sigma - kb)$$

The left hand term represents the payoff to playing $\sigma$. The right hand is the payoff to deviating. As each player can move, it will match price, and thus in each round, the profits will be shared among the $\frac{1}{i+1}$ players at the lower price $(p - kb)$. After that (N rounds), all players will be at $(p^\sigma - kb)$ forever.

We can simplify this to obtain:

$$p^\sigma \geq (1-\delta) \sum_{i=0}^{N-1} \frac{(p^\sigma - kb)N}{(i+1)} \delta^i + \delta^N (p^\sigma - kb) \tag{4.10}$$

For notational simplicity, we define:

$$\phi = \sum_{i=0}^{N-1} \frac{\delta^i}{(i+1)}$$

Note that this is the sum of the first $N$ terms of the Harmonic series with an increased discount term each period. We can restate Eqn.(4.10) as:

$$p^\sigma \geq (1-\delta)\phi(p^\sigma - kb)N + \delta^N (p^\sigma - kb) \tag{4.11}$$

Solving for $p^\sigma$ yields:

$$p^\sigma \leq \frac{kb(-N\phi + N\phi\delta - \delta^N)}{1 - N\phi + N\phi\delta - \delta^N} \tag{4.12}$$

□

From this Theorem, we can easily derive our two conclusions:

**Theorem 8.** *In the **asynchronous** RIRG, if players play strategies in PP, the value of $\bar{p}$ varies with the parameters $(\delta, b, N)$ in the same manner as the optimal price matching price. That is: $\frac{\partial \bar{p}}{\partial \delta} < 0$, $\frac{\partial \bar{p}}{\partial b} > 0$ , $\frac{\partial \bar{p}}{\partial D} > 0$ , and $\frac{\partial \bar{p}}{\partial N} < 0$.*

and

**Theorem 9.** *For an instance of the* ***asynchronous*** *RIRG with N players playing strategies in PP,* $\forall\ \epsilon > 0$ *there exists a tuple* $(\delta, b)$ *such that* $p_m < c + \epsilon$ *where* $p_m$ *is the price realized in the market.*

### 4.5.3 FPSS-Like Assumptions

The RIRG assumption of splittable flows and a single destination maps to a variety of practical contexts. For example, many ISPs offer a single price for all Internet routes, making "the Internet" the single destination. Further, while flows over BGP are confluent, routing technologies such as multihoming and overlays enable customers to split traffic among the providers. This split can be done in time or by selecting some granularity smaller than the destination advertised by the ISP.

However, it is useful to relax both of these assumptions. Note that if we only relax the assumption of a single destination and now allow multiple destinations, we now just have multiple copies of the same single link game. Conversely, if we relax just the assumption of splittable flows but still have only one destination (and break ties in a constant and deterministic fashion) then we have standard Bertrand competition where the only equilibrium is *price* = *cost* and thus no repeated equilibria.[11]

We formally define the new game below. In summary there are three key differences as compared to the RIRG:

1. Flows are confluent
2. There are two destinations
3. A second-price auction sets the allocation and prices

These three relaxations map directly to the FPSS model and our counter-example from Section 4.2. Consider, for example, the limited topology depicted in Fig. 4-5 with two players. As in FPSS, each provider advertises a single bid for its network. Despite the changes from the RIRG, the players face a similar decision: bid low to to obtain more traffic or concede one link to the other player in a repeated equilibrium. We define this game formally as a game among $N$ players below. Each player $P_i$ will be the low cost provider to some destination $t_i$.

We define the game formally as:

**Model: $N$-Player Repeated VCG Routing Game**

[11] Here we assume that time-dependent or such alternating period strategies are implausible.

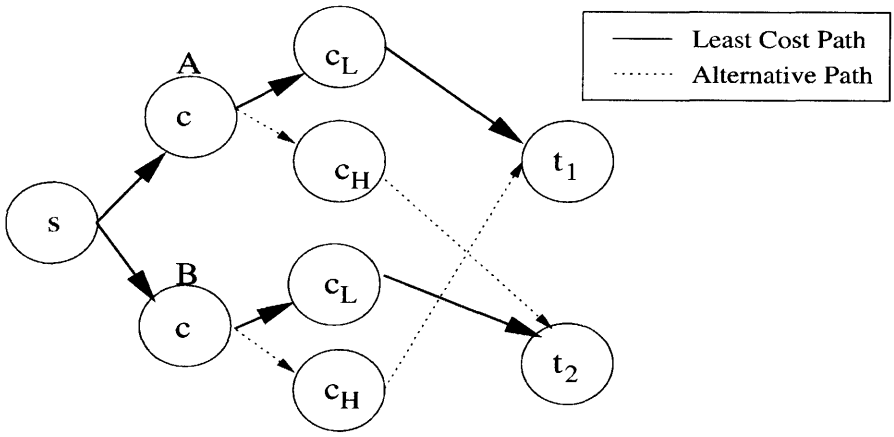



Figure 4-5: The $N$-Player Repeated VCG Routing Game with $N = 2$. With $c_H > c_L$, A is on the LCP to $t_1$ whereas B is on the LCP to $t_2$.

- There is only one source and N destinations, with $\frac{T}{N}$ units of flow from $s$ to each of $t_1$, ..., $t_N$.
- There are $N$ networks in the game ($P_1$, ..., $P_N$), each connected to the source. These $N$ networks have identical cost $c$ and all paths have equal quality.
- Between each pair $(P_i, t_i)$ are other networks providing connectivity. Each network has a fixed price, $c_h$ or $c_l$ where $c_h > c_l$. Each $P_i$ is connected to $t_i$ via a network with cost $c_l$ and connected to $t_j$, $j \neq i$, via a network with cost $c_h$.
- Each AS is infinitely patient with respect to the time value of money.
- All bids are common knowledge as in the prior game.
- $\delta$ models the finite but unknown duration

**Play of the Game**

1. The game proceeds in a series of rounds, each of length $d$, a constant that is common knowledge.
2. At the start of each round, each of the players advertises a *single bid* simultaneously. This value represents a (perhaps truthful) per-packet cost.
3. For the entire period, for each destination, traffic is routed over the provider with the lowest bid. In the event of a tie, traffic is sent to the lexicographic first network. Thus, all flows are confluent.

4. Each provider is paid for the number of packets that transit its network. The price per packet is set by the (second-price) VCG mechanism.

We define the critical price here in a similar fashion to the prior game:

$$p^* = \max_p \text{ such that } \pi_i(p,p) \geq (1-\delta)\pi_i(p-x,p) + \delta\pi_i(p-x,p-x)$$

for a given tuple $(\delta, \pi)$ and any $x \geq b$. However, note that the profit function, $\pi(.)$, and selection of $p^*$ is more subtle than before:

- The single bid requirement forces the minimum profitable deviation to be larger than the minimum bid size. For example, if two players are at a given $p$ and P1 decreases its price by some $\epsilon < c_h - c_l$, there will be no benefit to this deviation.
- The deterministic tie-breaking causes an asymmetry. A player later in the lexicographic ordering must exhibit a (slightly) larger price decrease to gain the additional traffic.
- The nature of the second price auction is that if $(p-x)$ is the lowest bid, then $\pi_i(p-x,p) = Tp$ not $T(p-x)$ as in the first-price auction.

Despite these differences, we are able to obtain a similar result for the protocol period, namely that $\frac{\partial p}{\partial \delta} > 0$. (We defer this proof to the appendix.)

**Theorem 10.** *In the Two-Player Repeated VCG Routing Game,* $\frac{\partial p}{\partial \delta} > 0$.

*Proof.* Deferred to Appendix □

While the impact of $\delta$ is the same in the 1st and 2nd price mechanisms, the impact of $b$ is different. First, unlike the first price auction, a player must deviate by an amount larger than the minimum bid size to gain additional profit. Therefore, the minimum bid size is not relevant in determining equilibrium price (beyond rounding). This can be viewed as a positive or negative. In the repeated first-price mechanism, as $\delta \to 0$, $p^I \to b\frac{N}{N+1} + c$ (Theorem 2). However, in the repeated second price mechanism, as $\delta \to 0$, $p^{II} \to y(N-1) + c$ (Lemma 2 in Appendix). Both prices correspond to the maximal values of the set of undominated strategies in the one-shot game. Thus, from an implementation perspective, while it is possible to force $p^I \approx c$ independent of topology, $p^{II}$ may be significantly bound away from $c$ even with the slightest possibility of repetition. These elevated prices, or more simply this

lack of control, can be viewed as another weakness of the VCG mechanism in the repeated game.[12]

### 4.5.4 Heterogeneous Costs

Above we saw that cooperation is possible in spite of – and in fact facilitated by – heterogeneous cost structures. Examining heterogeneous costs provides better insight into the strengths and weaknesses of our result.

With heterogeneous costs repeated can and still *may* exist. Let $p^*(c)$ represent the $p^*$ in a game with homogeneous costs of $c$. In the case where we have $c_1 < c_2$, P1 has the choice of i) selecting a repeated equilibrium in which the market is split or ii) pricing below $c_2$ and taking the whole market. This corresponds to bidding $p^*(c_1)$ or $c_2 - b$ respectively. For example, if we have $c_1 = 1$, $c_2 = 1.1$, and $p^*(1) = 3$, then we would expect the equilibrium price to be 3, even though P1 could undercut and price at say 1.09.[13] This logic can be generalized to derive an equilibrium price in the case of heterogeneous costs. Given this price, the results from prior sections – the impact of $\delta$ and $b$ in the first price auction and the impact of $\delta$ in the second price auction – still hold. Further, implicit in the second example is that differences in costs can further enable repeated equilibria in practice as it may cause one firm to effectively cede a market to a competitor in exchange for another market (e.g., domestic vs international).

This example also underlines a lesson for protocol designers. With the assumption of heterogeneous costs, it is possible to construct examples where the repeated outcome is the same as the static outcome and the protocol period and field width are not of great importance. However, there still exists a large class of instances where the repeated strategy will be the relevant one. Because one can rarely be sure about the network on which a designed protocol will be run, consideration of results presented in this chapter are therefore important.

## 4.6 Discussion and Future Work

The key conclusion of our work is that ***basic properties of the underlying protocol can have a significant impact on the equilibrium price.*** We have shown

[12]This is in addition to the reasons outlined in [84] which relate to information revelation. In this game, all information is common knowledge.

[13]Note that in this case $3 < p^*(c_2)$. So the profit to P2 is less than if costs were homogeneous.

this conclusion to be robust to multiple assumptions and practical conditions. This leads us to several interesting conclusions and observations.

### 4.6.1 Impact of this Work

One impact of this work is that we have endowed the protocol designer with a set of new tools, perhaps previously hidden. For example, in a simple first-price setting we can achieve lower prices through a longer period (increase $\delta$), a wider price field in the protocol (smaller $b$), and/or a less granular bandwidth using (smaller effective $b$). Holding other concerns aside, this means a consumer who has control over the protocol and seeks to limit price may find these useful.

However, this power also exposes new and unavoidable questions for mechanism and market selection. In particular, the tools are a double-edged sword as they raise questions of what the designer *should* do. For example, the interests consumers and suppliers may be at odds with each other. This could induce protocol alterations or issues of market and mechanism selection. This is another example of a "tussle" [17]. Further troubling is that these parameters are fundamental and thus *unavoidable*. Removing period restrictions creates a period implicitly defined by the players' reaction time. Likewise, in any networked protocol, there is a maximum level of granularity. This poses interesting questions regarding the possibility of flexible and/or self-adapting protocols and frameworks.

This insight speaks to the importance and practical nature of repeated games as a tool. In this chapter, the repeated model exposes the inherent limitation of the FPSS model, presents the fundamental problem of the repeated equilibria, and provides us with a means of addressing this problem through practical measures. As such, the positive results curtail the initial negative results and present repeated games as a practical tool.

### 4.6.2 Future Work

There are several assumptions in our models that should be relaxed and further examined in future work. We present them below, grouped by high level themes. We believe that with these relaxed assumptions, repeated equilibria can arise and when they do the relationships we find will still hold. Nonetheless, there is room here for considerable future work, which may provide insight into the other system parameters.

One area of generalization involves the consumers. In the model of this chapter, traffic (as in the FPSS model) is fixed and exogenous. A first question would be how to model a more active consumer – as a single user (e.g., an ISP or a CDN) or a number of separate users. Clearly at a particular interchange there is only one user. But it is likely that firms competing for one user will be competing for a number of similar users. Given an active user, we must now consider what else the user can consider and do. One obvious extension is to include some elasticity of demand function. Another angle would be to consider a model where there are a small number of users, which might permit a larger class of strategies for the users, such as selecting the firm that prices lower for an extended period of time (not just one round).

Another extension to the user and network models is quality of service. In our model, there is one fixed level of performance which does not play a factor in the game. A simple way to incorporate performance into the game is to give each firm a QoS type and introduce some notion of QoS into the user's utility function. To the extent that QoS were static and users homogeneous, it is unclear that this would qualitatively change the game. A firm with a worse QoS type would effectively have a higher cost, depending on the user's relative ranking of these factors. However, if QoS were negatively correlated with load, this could potentially create some interesting dynamics, as lowering price would increase load thereby decreasing quality and potentially causing the deviating firm to obtain less traffic than it would have in the simpler model. Such a model could also incorporate capacity constraints – which could either be fixed (representing a medium term horizon) or variable with build-out decisions (representing a much larger meta-game). In both, quality would presumably be a function of load relative to capacity. Results from such congestion-based models would have strong parallels to the recent results on the Price of Anarchy in networks [86] [53].

Focusing just on the networks, other extensions are possible. In our model, per-unit costs are fixed. Another choice would be model cost as a function of load, perhaps with a diminishing per-unit cost as volume increases. Another extension would be to consider costs which decrease over time. In either case, as costs decrease, one would expect some of that savings to enable more competitive pricing. This raises a question, in a game with imperfect information, as to whether or not price decreases are due to deviations or just natural forces (i.e., lower equilibria). Another way to create similar uncertainty would be to allow traffic volumes to fluctuate over time and/or to not have the winning bid of the prior round common knowledge in the next. We could also consider more general network topologies with multiple active

players along each path. Here we effectively have a chain of games being played. In addition to adding another dimension of complexity to analyze, this also can present some uncertainty in the input prices to the networks (since now cost is a function of the network's internal cost as well as its upstreams' prices). As discussed in the prior paragraph, another approach is to model cost as having two parts: a fixed component that is a function of capacity and a variable component based on load (to capture operational expenses).

Another particularly interesting area for future work is in the area of equilibrium notions, as indeed the question of appropriate equilibrium notions for repeated games is an open question [40]. In this chapter, we user the notion of sub-game perfection. While the SPE concept provides us with some useful properties, it is not sufficiently limiting as it still permits a large class of strategies that do not seem reasonable – such as the grim strategy of setting price to cost in response to a defection.

While it is unclear that there is a unique strategy which should be correct for this game, there are some clear properties. In steady state, a small number of sufficiently patient players should be able to maintain price above cost under reasonable circumstances. If and when a player deviates to some $p^* - x$, the other players should react by matching price, or perhaps lowering price even further. And over time, prices may rise and may or may not reach $p^*$ once again. These are the properties that are embodied in the definition of proportional punishment strategies in Section 4.5.1. Unfortunately, no equilibrium notion corresponds to this definition or that of the simpler price matching strategy. (We discuss this further in Section 6.4.1.)

There are numerous alternative equilibrium notions to consider. For example, Pareto-perfect and negotiation-proof equilibria encapsulate the notion that players are likely to select equilibria which are somehow (weakly) better for all players. As such, they do not permit the grim strategy. However, they do not permit a real punishment phase as such phases are clearly inefficient. On the other end, concepts such as evolutionary stability are not likely to be useful, as a defecting strategy will outperform a cooperative strategy. (This corresponds to the notion that a single foolish or myopic player can ruin an oligopoly.)

One area of this chapter that does provide some assistance is the analysis of the asynchronous model in Section 4.5.2. By making play asynchronous, we limit the scope of credible threats. For example, setting price to cost in response to a defection is not credible as it is strictly better to simply reduce price by the smallest amount possible and get the entire market share. However, this model still permits a large class of punishments. The problem here is that the punishment of reducing price

in every period (by $b$) until it reaches $c$ is still SPE. While this strategy is clearly Pareto dominated by any stationary cooperative strategy, the reduce-in-every-period strategy can be used as a threat to increase the equilibrium $p^*$. While it is certainly possible to add assumptions and requirements to further limit the equilibria, we have not yet found a set of well-founded and defensible assumptions that achieve this goal.

## 4.7 Summary

In this chapter we develop a model of incentive (or price) based routing that captures the notion of repetition, which is a vital aspect of practical applications. We see that the FPSS result does not directly hold here since it is not strategyproof in the repeated game. For a simple general model we are able to show that while prices can increase in general settings, their value is tied closely to certain, seemingly benign, properties of the underlying protocol. As such, we see that the protocol designer has greater control on the market than otherwise realized. We also show that these conclusions hold in more general settings, such as asynchronous play, a 2nd price auction, the case of multiple destinations, and the case of heterogeneous costs. Taken together, these results present an interesting and novel relationship between routing protocol design and economic considerations of practical importance.

## 4.8 Proofs

### 4.8.1 Proofs from Section 5

**Proof of Theorem 3**

*The protocol period and the market price are positively correlated – or* $\frac{\partial p}{\partial \delta} > 0$.

*Proof.* From equation (4.7) we have:

$$\frac{\partial p}{\partial \delta} = \frac{b}{(-1+\delta)(-N+N\delta-\delta+1)} \tag{4.13}$$

We seek to show that this ratio is positive. Clearly, the numerator $b > 0$ and the first term of the denominator $(-1+\delta) < 0$.

Considering the other term of the denominator, we have:

$$(-N+N\delta-\delta+1) = N(\delta-1)+(1-\delta)$$

$$= (1-\delta) - N(1-\delta) = (1-\delta)(1-N)$$

Since $(1-\delta) > 0$ and $(1-N) < 0$, we have $(-N+N\delta-\delta+1) < 0$. Thus,

$$\frac{\partial p}{\partial \delta} = \frac{+}{(-)(-)} > 0$$

as desired. □

### 4.8.2 Proofs from Section 6

**Proof of Theorem 4**

*Proof.* Since $\sigma$ is a SPE, we know that the one-stage deviation property must hold for every history and time step. Therefore, we examine a given strategy at a given decision point.

For any $\sigma$ cooperating in a period yields $\frac{p_\sigma^t - c}{N}$ whereas deviating yields $p_\sigma^t - c$. Therefore, the benefit of deviating is:

$$B = (p_\sigma^t - b) - \frac{p_\sigma^t}{N}$$

The cost to this one-stage deviation is:

$$C = \delta \sum_{t>t_0} \delta^t \left(\sigma_h^t - \sigma_{\hat{h}}^t\right)$$

Since $\sigma_h^t - \sigma_{\hat{h}}^t < kb$, we therefore have:

$$C \leq \frac{\delta kb}{(1-\delta)}$$

The one-stage deviation property holds iff $B \leq C$ or:

$$(p_\sigma^t - b) - \frac{p_\sigma^t}{N} \leq \frac{\delta kb}{(1-\delta)}$$

We can rewrite this in standard form as:

$$p \geq (p-b)N(1-\delta) + \delta(p-kb)$$

which we can solve to yield:

$$p \leq \frac{b(\delta N - N - \delta k)}{(\delta - 1)(N-1)}$$

Further, if $\sigma_h^t - \sigma_{\hat{h}}^t = kb$, then this expression holds with equality. □

**Proof of Theorem 2** *For an instance of the RIRG with N players playing strategies in PP, there exists a tuple* $(\delta, b)$ *such that* $p_m < c + \epsilon$.

*Proof.* From Theorem 4 we have that the highest possible price in $PP_k$ is given by the strategy which punishes by a factor of $k$ in each period which is in turn bound by:

$$p_m \leq \frac{b(\delta N - N - \delta k)}{(\delta - 1)(N-1)} + c \tag{4.14}$$

This bound also holds for any equilibrium even when players play different SPEs as any punishment weaker than consistently punishing by k will only decrease the market price.

Therefore if we seek to have $p_m < c + \epsilon$, we can set:

$$\epsilon > \frac{b(\delta N - N - \delta k)}{(\delta - 1)(N - 1)} \tag{4.15}$$

which can be readily solved.

In particular, for a fixed $b$ we have:

$$\delta < \frac{-\epsilon N + \epsilon + bN}{-\epsilon N + \epsilon + bN - bk} \tag{4.16}$$

and for a fixed $\delta$ we have:

$$b > \frac{\epsilon(\delta - 1)(N - 1)}{\delta N - N - \delta k} \tag{4.17}$$

□

### 4.8.3 Proofs from Section 7

**Lemma 2.** *In the N-Player Repeated VCG Routing Game, the equilibrium bid is given by:*

$$p^* = \frac{yN(2N + \delta N - 2\delta - 1) + \delta(Nc - y - c) - y + c}{(1 - \delta)(1 - N)}$$

First, we must define the player's profit function. Since the first player (lexiographically) needs to make the smallest deviation, we focus our analysis on this player, without loss of generality. In particular, since the players are homogenous, if some other player (which must make a larger sacrifice) will deviate, then the first player will as well.

Define $b_{(i,j)}$ to be the sum of the bids on the path to $j$ via $i$. That is,

$$b_{(i,j)} = \begin{cases} p_i + c_h, & i \neq j \\ p_i + c_l, & i = j \end{cases}$$

Next we define the indicator function, $I$, which denotes if $i$ is on the LCP:

$$I(j, p_i, p_{-i}) = \begin{cases} 1, & b_{(i,j)} = \min_k b_{(k,j)} \\ 0, & \text{otherwise} \end{cases}$$

Define $\beta(j)$ to be the $k$ such that $b_{(k,j)}$ is the second-least (or tied for the first-least).

Define $y = c_h - c_l$.

We can now expand $\pi(.)$ based on its definition:

$$\pi_i(p_i, p_{-i}) = \sum_{j \neq i} I(j, p_i, p_{-i}) \frac{T}{N}(\beta(j) - y - c) + I(i, p_i, p_{-i}) \frac{T}{N}(\beta(i) + y - c)$$

The first term represents the sum of all profits derived from being on the least cost path to some $t_j$ $j \neq i$. The second term represents the profits to being on the LCP to $t_i$. These expressions are derived directly from the VCG calculation. Note that per the VCG, $p_i$ does not appear in the profit function, except for being in the indictor term, $I(.)$.

We now turn to the question of the equilibrium conditions. We know that in equilibrium, bidding $p$ must be better than deviating by some $x$, or:

$$\frac{T}{N}(p + y - c) \geq (1 - \delta)T(p - (N - 2)y - c) + \frac{T}{N}\delta(p - x - c)$$

where $x \geq y$.

Since each side is monotonic in p, we consider only the case where $x = y$ to yield:

$$(p + y - c) \geq (1 - \delta)N(p - (N - 2)y - c) + \delta(p - y - c)$$

We can solve to obtain:

$$p \leq \frac{yN(2N + \delta N - 2\delta - 1) + \delta(Nc - y - c) - y + c}{(1 - \delta)(1 - N)} \tag{4.18}$$

Thus, in equilibrium, we have:

$$p^* = \frac{yN(2N + \delta N - 2\delta - 1) + \delta(Nc - y - c) - y + c}{(1 - \delta)(1 - N)} \tag{4.19}$$

And the VCG price, $p_V$ is given by:

$$p_V = \frac{yN(2N + \delta N - 2\delta - 1) + \delta(Nc - y - c) - y + c}{(1 - \delta)(1 - N)} + y \tag{4.20}$$

**Proof of Theorem 10**

*Proof.* We take the partial derivative from Eqn(4.20), which simplifies to:

$$\frac{\partial p_V}{\partial \delta} = \frac{2y}{(\delta - 1)^2(N - 1)} \tag{4.21}$$

Since $N \geq 2$ and $\delta < 1$, we have:

$$\frac{\partial p_V}{\partial \delta} = \frac{(+)}{(-)^2(+)} > 0 \tag{4.22}$$

□

# Chapter 5

# Repeated-Game Modeling of Multicast Overlays

This chapter studies multicast application overlay networks in a repeated-game framework. In these overlays, users have both the motivation and the means to alter their position in the overlay tree. We introduce a repeated-game model of user behavior that captures the practical tradeoff between a user's short-term desire for quality and long-term desire for the network's continued existence. We simulate overlay tree-formation protocols with this model to study their robustness to selfish users. We show that this model can explain user cooperation and provide insight into how overlay systems scale in the absence of heavyweight mechanisms or identity systems. We also use the model to derive practical guidance on how to make multicast overlay protocols more robust to selfish users.

## 5.1 Introduction

The benefits of IP multicast have been long discussed and documented. By creating copies of the data within the network instead of at the source, IP multicast simultaneously accomplishes several design goals to provide a scalable infrastructure for wide-scale distribution of real-time data. Canonical examples include low-bandwidth applications such as stock tickers and high-bandwidth applications such as live streaming video.

However, IP Multicast has not been significantly deployed on the public Internet. Several well-documented and well-understood architectures, protocols, and tools exist

[23, 28, 101, 4, 88] and are implemented in many routers [50, 55]. Nonetheless, there are many technical and economic reasons why these have not been adopted [26], including a lack of clear economic incentives for ISPs. While proposals to address the economic and [33] and technical problems exist, many challenges remain.

An alternative approach to realizing many of the same design goals is *application overlay multicast*. In an application overlay multicast network, end nodes use IP unicast to create a tree representing a virtual multicast network. Such overlays have been examined extensively in the literature and are now beginning to be deployed.

By relying on end-user systems, the overlay eliminates the need for ISP deployment but, like other peer-to-peer applications, encounters the issue of user-incentives. In practice, the location of a node in the tree may define its quality of service. Particularly because the system relies on end-user machines and connectivity, nodes deeper in the tree may suffer from increased latency, jitter, or loss. Further, nodes supporting many children may suffer from increased bandwidth or CPU utilization. This may cause nodes to move higher in the tree and/or support fewer (if any) children – to the extent that they can do so. Even this simple selfish (not malicious) behavior can wreak havoc on the system and its efficiency. Therefore, a proper system design must consider these incentive concerns.

One approach to these incentive issues is to design mechanisms to detect and/or prevent cheating directly. Proposed tools here include identities, monitoring systems, micro-payments, and reputation schemes. Indeed, a study of the system using standard one-shot game theory suggests that we cannot engender user cooperation without either financial incentives or tight monitoring. However, it is often impossible to implement either payment schemes or tight controls; even if possible, they likely require a heavy infrastructure investment. Further, systems (e.g., [14]) *without* this additional infrastructure often perform reasonably well in practice!

In this chapter, we demonstrate that this apparent paradox can be explained by the key observation that *even selfish users want the system to exist in the future*. We use the tools of *repeated game theory* to develop a model of user behavior that captures this positive aspect of user incentives. In our model, a user's benefit depends on her position in the overlay tree. A strategic user evaluates each potential action based on its total expected benefit: the sum of her immediate benefit and a (discounted) future benefit over the expected lifetime of the system. An action to deviate from the protocol might improve her immediate benefit, but have the side-effect of degrading the system performance and hence shortening its expected lifespan. Thus, she will deviate from the protocol only if the immediate gains from cheating exceed the future

expected loss.

In contrast to most earlier work on mechanism design for networks, we assume a *minimal infrastructure*. We explicitly assume that there is no central trusted entity, nor any form of a payment system. Further, we do not attempt to devise an effective monitoring, identity, or reputation system. These assumptions reflect the reality of the current Internet. Further, even if such machinery could exist, it would likely come at a significant (monetary or performance) cost. It is therefore important to consider alternative design approaches when possible. With this model, we study the performance of overlay multicast formation protocols in the face of selfish user behavior. We aim to evaluate the protocol performance, and to identify protocol parameters or features that improve robustness to selfish users.

Another key difference of our model is that the repeated-game analysis makes the motivation for cooperation *endogenous* to the model. Whether a user deviates from the protocol is determined by her current and future payoffs, and is thus directly dependent on the system design. In contrast, earlier explanations of cooperation in overlay networks assume that users have an exogenous fixed *type* that determines their propensity to cheat. For example, users are modeled as cheaters or non-cheaters [67], or assigned an altruism parameter [47] [37]. User behavior is probably influenced by exogenous type differences as well as other factors such as bounded rationality. An ideal model would include all these factors. We however believe that an endogenous model of cooperation is more useful for two reasons. First, it has fewer degrees of freedom than models requiring a distribution of altruism types; this makes it easier to make sharp, testable predictions. For example, our model naturally predicts that, as the number of users grows, the fraction of users who deviate will tend to increase. Such a trend was earlier observed in various systems by Huang *et al.* [48]. Second, even if real user behavior involves both a system-independent altruism type and a system-dependent endogenous incentive to cooperate, the latter aspect is likely to be more useful to system designers.

We believe these two differences are significant and use our model to derive novel practical insight into multicast overlay formation techniques. Because our model explains a user's willingness to cheat based on her location in the tree, we can compare the robustness of different topologies and tree formation techniques. In simulation, we compare the tree-formation protocol NICE [6] to a naive tree-formation algorithm that maximizes the network efficiency. The results suggest that NICE is more robust to selfish users, and can actually lead to greater efficiency under fairly general conditions. We then consider the impact of a basic NICE parameter, the cluster size. We find

that, under reasonable conditions, increasing the cluster size can create more robust, and thus potentially more efficient, trees.

Depending on interests and background, the reader may benefit from a non-linear read the first time through this chapter. In Section 5.2, we present some background information on overlay trees, focusing on the NICE protocol. In Section 5.3, we explain and motivate the problem. We follow this in Section 5.4 by explaining why the problem cannot be readily solved with a clever protocol and proving a formal statement of hardness for the one-shot game. (This section could be skipped during a first read.) Given this motivation, we present our model in Section 5.5 and our simulator in Section 5.6. In Section 5.7 we present our core results. Because we use a simpler model at first to help clarify the intuition, we then consider a wide-range of relaxing assumptions in Section 5.8 and see that our results are insensitive to these assumptions. (This section can also be skipped on a first read if desired.) We then finish with a summary and conclusions in Section 5.9.

## 5.2 NICE and Multicast Tree Formation Protocols

The goal of a tree formation protocol is to create an efficient overlay tree connecting user-nodes to a source. The notion of efficiency however is multi-variate. Important dimensions include scalability, communication overhead, and convergence time in the face of nodes joining and exiting the system. Performance metrics that are considered include *stress*, the maximum load induced on any router in the network, and *stretch* or relative-delay-penalty (RDP), the ratio of the latency of receiving a packet through the overlay versus directly from the source.

Due to differing requirements and the tradeoffs between these metrics, there are a variety of proposed protocols for this particular overlay problem. First, there are centralized protocols ([78, 83]) with a single control node. The distributed algorithms are typically discussed in two groups. The first is mesh-first protocols (e.g., Naranda [15]) which build a mesh between the nodes to allow for multiple sources and greater reliability. The second category is tree-first protocols, which construct a specific tree for a given source and set of nodes. Within this class are protocols that construct trees with specific algorithmic properties [6, 60, 104]. Another class of tree-first protocols allows much more free-range to the nodes, essentially allowing them to connect in the tree to the node who best serves them. [14, 52]

For most of this chapter, we examine a particular tree-first protocol, NICE. Tree-first protocols are more appropriate in single-source applications such as the ones we

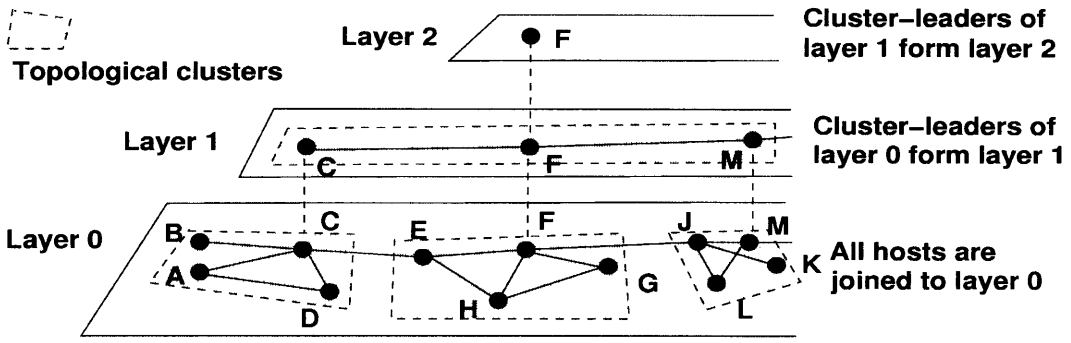



Figure 5-1: A three-dimensional depiction of the NICE protocol. Leaders from the lower layers become the members of the higher layers. (Figure taken from [6])

consider. Among the tree-first protocols, those that do not specify strong algorithmic constraints easily facilitate strategic manipulation. (For example, if a node can select its number of children, as in [14], it is easy to select 0.) Finally, among tree-based and mesh-based protocols, NICE has been shown to have good performance across a variety of metrics [6] including communication overhead. While some of these metrics are orthogonal to our study, it makes NICE a good candidate for exploration.

NICE forms a hierarchical tree of clusters. Each cluster contains nodes that are close to each other, according to some cost metric. These nodes in turn select the centroid of the cluster to be their leader. Clusters are arranged in a hierarchy of layers, such that the members of a cluster at layer $\ell$ are leaders of clusters at layer $\ell - 1$, as depicted in Figure 5-1. As such, all nodes belong to a layer 0 cluster.

The tree maintenance is a completely distributed process. A node joins the tree by descending through the tree, selecting the closest node in a given cluster and then querying it for its children. To balance the tradeoff between stretch and stress, the tree maintains a cluster size parameter, $k$. If the size of a cluster falls below $k$ it is merged with a nearby cluster that shares a common parent. Similarly, if a cluster becomes larger than $3k$, the cluster is split with the new cluster leaders as children of the old cluster leader's parent. The cluster leaders also periodically attempt to improve the tree, either by transferring leadership to a new centroid or by finding a better parent to which it should attach.

Further detail and simulation results for NICE are presented in [6].

## 5.3 The Problem

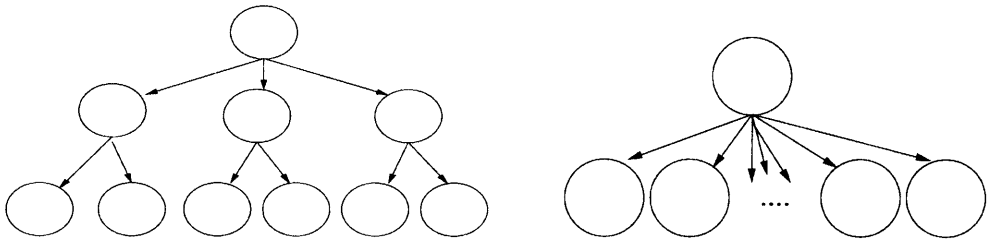
A crucial problem with the tree formation protocols is that they assume that users will be faithful to the protocol. Unfortunately, users have *both the motive and the means* to alter their location in the tree. The location in the tree can significantly impact the stream quality and the load on a node. Further, in all of these protocols, there are a number of ways to lie and thus alter one's position in the tree. In this section we discuss these motives and means and examine some alternative approaches to the problem. Throughout this discussion, we refer to this tree altering behavior as *cheating.*

### 5.3.1 Why Nodes Want to Cheat

The utility of an end-user is a function of 1) the content (data, streaming video, etc) obtained from the tree; and 2) the responsibilities incurred from participating in the tree. These are in turn defined user's location in the tree and the tree structure. Consequently, the tree structure is very important.

For a large class of applications, *a node's proximity to the root significantly impacts the user utility.* In any networked application, processing by several intermediate nodes creates the potential for decreased quality – including increased latency, loss, or jitter. This is especially relevant in the case of overlay networks where the intermediate nodes are assumed to be end-user machines, with potentially limited resources (such as bandwidth or processing power). Further, traffic on overlay networks must travel longer distances through the underlying physical network (i.e., stretch $> 1$). This further increases the chances of quality degradation. The impact of this degradation (e.g., increased loss, latency, and jitter) can be significant to end-user experience. For example, loss (and to some degree jitter) can be particularly harmful for streaming applications and latency is a highly relevant metric in event and information dissemination networks.

Another practical consideration is that *supporting children has a negative impact on a user's utility.* Each additional child supported requires overhead state and resources. While this overhead need not for example scale linearly with the number of children, its impact can be non-trivial. This can be problematic for several reasons: 1) the user may be running other tasks at that time (e.g., email, word processing), 2) the underlying application may already be CPU intensive (e.g., streaming), and 3) the hardware and operating system of the machine is likely not optimized to function as a server. Even if the user has sufficient resources, the additional load can still



(a) Efficient Topology (b) Selfish Users

Figure 5-2: Selfish Users Lead to a Different Topology

decrease the user's experience. For example, the user may wish to have such a buffer of resources available on demand (e.g., if she wishes to open another video, launch another CPU-intensive application). Practical validation of these concerns can be seen in the fact that several peer-to-peer and application multicast overlay networks allow users to disable being a parent node [14] [64].

These two assumptions, that the proximity and that the number of children impact the user's utility, are reasonable and mild. We are not requiring that utility be strictly decreasing in both of these parameters, nor that the relationship must be dramatic. Indeed in Section 5.8.1, we consider a wide range of shapes for these relationships including step functions.

The result of these two considerations is however significant. In the absence of any enforcement mechanism or concern for the future, these two factors can transform efficient overlay trees into degenerate, unicast trees. This is depicted in Fig. 5-2. Instead of the hypothetical overlay topologies in Fig. 5-2a, sufficiently greedy users will produce Fig. 5-2b. This has obvious deleterious effects on the various metrics of system performance, and even the system's continued existence.

### 5.3.2 How Nodes Can Cheat in NICE

Given the incentive to cheat, it is important to understand how readily a node could eliminate children and/or move higher in the tree. Unfortunately, there are a variety of ways to do one and/or both in most protocols. In some protocols, such as those where the user selects the number of children she wishes to support (e.g., the CMU system [14]) this is trivial. In others, such as NICE, the process is more subtle but also quite easy.

For example[1]:

- A node desiring to be part of a layer $\ell$ cluster could just claim to be the leader of a layer $\ell - 1$ cluster, even if such a cluster did not exist.
- A node can manipulate the heartbeat messages so as to become leader of a given cluster, and then continue to manipulate while refusing to relinquish leadership. This could be repeated multiple times to move up the tree. (This is discussed in further detail in [67].)

To support fewer children, a node could:

- Simply refuse to serve its children or partially-serve them (e.g.,transmit a fraction of the packets).
- Delay or drop probes from new children
- When joining the tree, select a cluster where performance is marginally worse but where it is very unlikely to become the leader.

The existence of these cheats should not be viewed as a flaw of NICE. It is designed to operate with faithful users and thus has no protection whatsoever from any of these cheats. Further, some of the cheats (such as some of RTT manipulation) can potentially be addressed readily. Nonetheless, the number of cheats and the difficulty in preventing some is very concerning.

### 5.3.3 Generalized Cheating Techniques

As can be seen by the example of NICE, there are a variety of ways in which a selfish (but not malicious) user can cheat in these networks. The cheats vary based on the protocol and algorithm used. Nonetheless, they can be placed into three primary classes:

- *Explicitly Not Supporting Children*: This form of free-riding may be the simplest form of non-cooperative behavior. One way to do this is to accept children but not transmit content to them. In some protocols, such as [14], this behavior is explicitly permitted as nodes can declare that they will not support children.

[1]These cheats are not designed to be optimal or the easiest, but rather simple and likely effective. In Section 5.4 we discuss how such cheats may be stopped – but also how some cheats cannot be readily prevented.

- *Measurement Manipulation*: Most protocols rely on some form of measurement (RTT, quality, or otherwise) to make tree formation decisions. It is therefore tempting and possible to somehow manipulate these measurements. For example, with RTT, it is trivial to delay a response and with the use of a proxy or predicting future pings, it may even be possible to reduce measurements. Further, if the protocol requires sharing of information between peers, this data could be fabricated.

- *Sybil Attacks* Sybil attacks involve the creation of multiple identities to benefit the true user. In practice, Sybil attacks are a second order attacks to thwart architectures designed to address incentive problems. For example, in the case of NICE, a user could simply claim that she is a layer $\ell - 1$ leader and thus should join a layer $\ell$ cluster. One simple defense would be to require the user to present the IP addresses of the user's children and descendants. A Sybil attack here would allow the user to create fake descendants and thwart that defense.

  Sybil attacks are particularly potent inasmuch as *they cannot be readily detected or prevented with technology alone* given the existence of network artifacts such as Network Address Translation (NAT) boxes. NAT boxes, used appropriately, enable the sharing of a single IP address by multiple end-users, as depicted. To accomplish this, the NAT box assigns each device within the network its own IP address from a private IP space (10.*) valid only within the network. To the outside world, each flow appears to come from the IP address assigned to the NAT box, but on a different port number. When the IP packet is returned to the NAT box from the remote host, the NAT box uses the port number to map the flow to the local IP address and deliver the data. Given this architecture, it is impossible to know whether or not two flows came from the same end-user, nor is it (in general) possible to know the number of users behind a given NAT box. For this reason, NAT boxes are also employed to provide privacy. However, this provides a simple method for a user to fabricate nodes with the same IP address.

  The power of NAT boxes for Sybil attacks rests in the ability to thwart detection. Consider a monitoring system which could test nodes to see if they were receiving the stream. This would prevent a node from claiming an invalid IP address or another, non-involved machine, as a child. But since the fake nodes appear on the same machine as the valid node, they will not be detected. (This

is discussed further in Section 5.4.1.)[2]

There are several possible, but ultimately incomplete, defenses to Sybil attacks. One simple defense is to have each user solve a computational puzzle. On legitimate users behind a NAT, the load would be negligible. The hope however would be that if the multiple users were really one machine, the computational load would be excessive. However, in this application, this is unlikely to be an effective solution. First, the system must support a wide range of end-user machines, capabilities, and spare load and thus the puzzle must be mild. Secondly, the number of fake children required to effectively cheat may, in many cases, be quite small. Thus, a user with a slightly-above average system could readily circumvent this defense. Another approach would be to force the NAT to reveal its information in some form. This has the downside of exposing potentially private information in the case of real NATs. Other approaches that are discussed in the next section include identities and payment schemes.

## 5.4 Why This Problem is Hard

Unfortunately, the problem presented in the last section is difficult and perhaps impossible to solve given practical constraints. The goal here is to distribute the content to the end-users in a manner that minimizes network load while providing sufficiently good end-user quality. As is seen, achieving this goal in the face of selfish users requires some additional tools. Further, each solution to a facet of the problem must be robust to a number of counter-attacks, including including Sybil attacks. While several approaches exist, each has weaknesses in practice.

This section explains why this problem is hard by examining these cheating techniques and potential solutions. In the first part, we consider several reasonable proposals for addressing this problem and discuss the weaknesses of these solutions. In the second part of this section, we narrow the problem and formally prove a result of hardness. In particular, the argument that there is no effective solution from an architectural perspective is equivalent to arguing that there is no solution to the one-shot game. Taken together, these motivate alternative approaches and in particular a repeated game model.

[2]Note that NAT boxes are not the only tool that is this effective, but rather the simplest. The requirement is that the user has a piece of IP space where she can create multiple addressable users. Therefore, a user in charge of a block of IP space could perform the same attack without a NAT – or could combine the two approaches.

### 5.4.1 Potential Solutions and Practical Challenges

To address the cheating techniques presented in the previous section, there are a variety of defense mechanisms which have been proposed and in some cases implemented. We present several of them below. Our examination is neither exhaustive in the breadth of approaches nor the depth which which we consider each technique. Instead, our goal is to highlight the important categories and explain relevant details.

In considering the merits of each approach, we make two key assumptions. The first assumption is that only simple and lightweight architectures are appropriate. Clearly there are many approaches to content dissemination.[3] However, if an end-user overlay is being used, it is very likely that a simpler and more light-weight approach is suitable. The second assumption, per the previous discussion, is that Sybil attacks from a given IP address can not be detected nor prevented. In light of these two assumptions, we believe that while each of the following approaches has merit, none satisfactorily addresses this problem.

- *Micro-Payments* The most direct approach to the incentive problem is to introduce payments to the system. Proposals for such systems include taxation schemes [46] or leveraging the VCG mechanism [91]. This of course requires some form of a payment system, which likely would require significant infrastructure and additional tools such as identities. Not only does this not exist on the Internet today, but moreover such an approach is in contrast with the light-weight distributed design approach of the overlay network. Therefore, it is unlikely to be appropriate for this problem.

- *Bilateral Monitoring and Agreements* In some peer-to-peer applications, bilateral monitoring and rewards can be employed. As an example, in the area of peer-to-peer storage Samsra [21] uses the notion of a claim to require a node using storage on another node to supply commensurate storage to another node in the system. These claims can be probabilistically checked to ensure compliance. This approach has also been considered in file-sharing networks [8]. However, the multicast application differs from storage and many other applications in the fact that the content flow is *unidirectional*. The parent never receives any content from the child. Therefore, these approaches are not directly applicable.

  To address this, one could artificially induce multiple topologies to the network. For example, SplitStream [73] periodically regenerates the network and

[3] One approach is to avoid any incentive problem by using a Content Delivery Network. Here the source pays the CDN, who in turn provides and controls all the server nodes.

attempts to punish free-riders. Similarly, one could imagine multiple concurrent trees with the source alternating packets among the trees. However, such an approach is still vulnerable to Sybil attacks.[4] For example, in NICE, if a node is able to connect to the root via a Sybil attack, no node is able to report that it is cheating.

- *Monitoring* Direct monitoring of nodes and their behaviors is an attractive, but difficult proposition. First note that in general there is no way for one node on the Internet to monitor the actions of another. However, one could imagine a system in which the source or indeed every node in the tree performed a series of tests on other nodes. For example, a monitoring node could pose as a new node to understand the tree topology, or could periodically require each node to provide information or pass a series of tests. While many of these approaches could prove hard and/or costly, they ultimately can be thwarted with Sybil attacks. In particular, any node can claim to have any number of children and descendants at the same IP address. A monitoring node could potentially require a series of checks (e.g., supply the MD5 of some recent packet plus a nonce) but since the fake and real clients are running on the same machine, any test that a real node could pass can easily be handled by a fake node.

- *Identities* A more direct approach to stopping Sybil attacks is through the use of identities. There are several ways of building a robust identity scheme. In particular, the identities could be tied to a form of identification difficult to forge (e.g., require in-person presentation of a passport) or could simply be costly to obtain (e.g., must be purchased).

  These approaches present at least two challenges. First, the system must obtain and verify these identities. Second, the approach of requiring identities may be inappropriate for the application due to privacy concerns or simply the hassle relative to the value of the content.

- *Reputations* Reputation schemes combine aspects of monitoring and identities. Many reputation and trust-inference systems have been proposed for peer-to-peer systems [65, 56] and even multicast-overlay networks [62]. These approaches are challenged in at least two ways here. First, they introduce additional overhead and complexity to the system. Second, there is a significant

[4] In [73], the authors of the SplitStream paper state that their approaches "can all be potentially defeated" with a Sybil attack. They suggest the use of strong identities, which we discuss in this section.

tradeoff between efficiency and robustness when using the reputation information. For example, consider a large number of legitimate users behind a NAT box. Likely, the efficient solution is to form a subtree linked to the larger tree near (or at) the source. In this topology, they interact minimally with the other nodes and thus may have a low reputation score. If the tree is significantly altered to punish these nodes, this by definition is inefficient for the system. However, if they are not punished, the algorithm is vulnerable to a Sybil attack.

- *Randomization and Fixed Inefficiencies* In several of the above approaches, we face the problem that a Sybil attack is effective in thwarting the mechanism by creating a virtual sub-network or subtree. One drastic solution to this problem, for example, is to construct the tree in a purely random fashion, without consideration for efficiency or quality. While this could potentially eliminate the incentive for cheating, the efficiency losses (in terms of quality and load) could be significant. Similarly, one could consider fixed-inefficiencies – for example, a set of users must support $x$ external users. Here $x$ is small, this is likely to not impact users and thus be moot. If however $x$ is large then we have introduced a large inefficiency into the system.

### 5.4.2 Hardness of a Solution for the One-Shot Game

The above discussion argues that many potential solutions for the problem of multicast overlay networks are not sufficient in the face of Sybil attacks; require overhead that may not be appropriate for this class of applications; and/or create tradeoffs between robustness and efficiency. We now seek to formalize this argument.

Our goal is to show that the problem, viewed as a one-shot game, cannot be solved. To do so, we formally define the game, the objective function, and the environment. We also assume that reputation and payment schemes do not exist, and formalize the argument about distinguishability for Sybil attacks. For clarity, we only consider deterministic algorithms. For this set of assumptions, we show that no tree formation algorithm can be efficient. **The key intuition is that an algorithm that effectively ignores Sybil attacks creates the potential for cheating (and thus an inefficient topology), whereas one that seeks to address it will directly create inefficiency.**

### Game Model

A fixed instance of the game is defined by:

- A network, $G = (V, E)$, $V$ and $E$ are finite.
- A set of nodes N to be served, $N \subset V$
- Each node is named by an IP address and port number
- A single source, $s \notin N$, $s \in V$ which sends a stream of data to all nodes directly connected to it. The source has infinite capacity and constant utility.
- A single, atomic, piece of content to be sent from the source to the end-users.

Given a game instance, an algorithm $A(G, N)$ produces a tree $T$, which connects the $N$ users to $s$ using edges in the network G. More formally, for a network $(V, E)$ and users $N$, $T = (V_T, E_T)$ such that $V_T = \{N \cup s\}$ and $\forall (i, j) \in E_T \; \exists \; \nu_{(i,j)} = ((i, y_0), (x_1, y_1), ..., (x_n, j))$ $s.t.$ $\nu_{(i,j)} \subseteq E$. We consider only deterministic algorithms.

Given a tree, each node's experience is captured by a utility function, which is defined as follows:

- $d_i(T)$ denotes the depth of node $i$ in rooted tree $T$ and $c_i(T)$ denotes the number of children that $i$ supports in $T$.
- $u_i(T) = u_i(d_i(T), c_i(T))$. $u(.)$ is strictly decreasing in both $d_i$ and $c_i$.

By omission, we are assuming that there is no payment mechanism nor reputation scheme.

### Assumptions about Indistinguishability

We now formalize the practical assumptions outlined in Section 5.4.1. For the purposes of this proof, we make the conservative assumption that users can only create fake users at the same IP address, as opposed to other IPs in the same space. Nonetheless, even this limited amount of cheating creates an insurmountable challenge for the tree formation protocol.

We define this problem as that of *indistinguishably.* If two networks, $G$ and $G'$ differ only in the fact that $G'$ has more nodes at a particular IP or set of IPs, in practice no algorithm can effectively disambiguate the two. To formalize this concept, we first define the set of possible disguising operations, which we call *fabrications.*

**Definition 2.** *Network $G'$ can be* fabricated *from network $G$ by node $i$ (denoted by $G \rightarrow_i G'$) iff $G'$ differs from $G$ only in that IP address $i$ has more nodes in $G'$ than in $G$.*

**Definition 3.** *The Potential Fabrication Set (PFS) of a network $G$, denoted as $G^*$, is defined as follows:*

- $G^0 = G$
- *For $i > 0$, $\hat{G} \in G^i$ iff $\hat{G} \in G^{i-1}$ or $\bar{G} \in G^{i-1}$, $\bar{G} \rightarrow_i \hat{G}$, and $i$ prefers $\hat{G}$ to $\bar{G}$*
- $G^* = G^\infty$

Since there are a finite number of addresses, we are guaranteed to have all possible profitable fabrications in $G^*$.

### Objective Function

With the notion of indistinguishably formally defined, it is possible to define a meaningful objective function. To do so, we will first introduce the notion of load *reduction* and use this to define the *benefit* of an algorithm. With this, we will define the objective function for the overlay tree construction algorithm.

Let $L(\ell, T, G)$ be the load on link $\ell \in E$ given tree $T$ and network $G$ and let $UNI$ represent the unicast topology over the true nodes, $N$, in the network.

**Definition 4.** *The* reduction *of an overlay tree, $T$,on a network, $G$, is defined as follows:*

$$R(T,G) = \left( \max_k \ s.t. \ \forall \ell \ L(\ell,T,G) \leq \max\left(\frac{L(\ell,UNI)}{k}, 1\right)\right) - 1$$

Conceptually, $R(T,G)$ is the minimum load reduction over all the edges in the network where the original load was at least 2. The motivation for the floor of 2 is that it in some cases a load of 1 may be the absolute minimum possible. [5]

As discussed, the $G$ observed by the algorithm, $A$, may not be the actual network but instead a fabrication of some $G'$. While $A$ may perform well on $G$, it may

[5]There are of course other metrics which may address this problem to some degree but we feel that this metric that we use most appropriately allows us to capture the motivation and intuition of the problem.

however perform poorly if it believes the network is $G$ when it is really $G'$. More formally, $R(A(G), G)$ may be high but $R(A(G), G')$ may be much lower. To capture this problem in our objective function, we define the actual *benefit* of an algorithm to be the minimum reduction obtained over the class of all graphs in $G^*$.

**Definition 5.** *The* benefit *of an algorithm on a network is defined as:*

$$B(A,G^*) = \min_{\hat{G} \in G^*} R(A(\hat{G}), G)$$

*if all $N$ nodes are served and 0 otherwise.*

While it may not be the goal of the algorithm to maximize the benefit per se, note that any worthwhile tree must satisfy at least three properties:

1. $B(A, G^*) > 1$ (and in most cases we would expect this to be significantly higher)
2. All $N$ nodes are connected by $T = A(G)$; and
3. All nodes satisfy the individual rationality constraint, that is, $u_i(d_i, c_i) \geq 0 \ \forall i$.

### Proof of Hardness

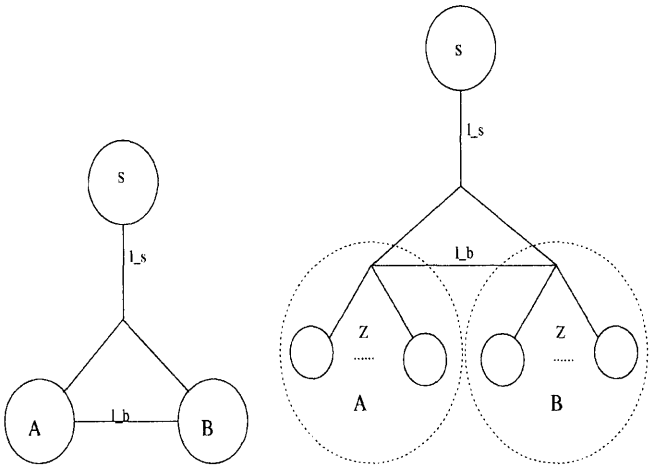
We now seek to show that this problem is hard. That is, there are simple networks where it is possible to create overlay trees with a high reduction value *but* given selfish users, the fabrication set is such that the benefit of any algorithm is 0.

**Theorem 11.** *For any algorithm, $A$, $\exists$ networks $G_0$ and $G_1$, $G_1 \in PFS(G_0)$, and utility function $u(.)$ such that:*

1. *There exists an algorithms, $D$, such that*
   *$R(D(G_0), G_0) > 1$ and $R(D(G_1), G_1) > 1$*
   *but*
2. *$B(A(G_0), G_0) = 0$ or $B(A(G_1), G_1) = 0$*

*Proof.* Consider the simple networks, $G_0$ and $G_1$ in Figures 5-3(a) and 5-3(b) respectively. The networks are the same except the two bottom nodes have $z$ children each in $G_1$.

We now define two simple overlay topologies. $T_0$ is any topology which contains a connection from $s$ to one or more nodes in A and one or more nodes in B. $T_1$ is any




(a) $G_0$ (b) $G_1$

Figure 5-3: Two indistinguishable topologies. No algorithm can determine if the $2z$ children in $G_1$ are legitimate or created solely to increase the utility of the real nodes.

topology which contains a connection from $s$ to one or more nodes in A *but no* nodes in B.

First, we show that there exists a $D$ to satisfy the first requirement of the theorem. For $G_0$ and $G_1$ (the true networks with faithful users), topologies $T_0$ and $T_1$ applied to $G_0$ and $G_1$ respectively satisfy the requirements for $D$. In the case of $G_0$ and $T_0$ we have a load of at most 1 per link as compared to a load of 2 with the unicast tree. In the case of $G_1$ and $T_1$ the only link with load greater than 1 is $\ell_1$. But $L(\ell_1, UNI) = 2z$. Thus, $R(D(G_1), G_1) = z > 1$.

The problem of course is when the networks are not faithful. Consider the case when the underlying network is really $G_0$, but A and B falsely claim to have $z$ children. Since this is exactly $G_1$, the output of $A$ will be as if the network were $G_1$ even though it is really $G_0$. Therefore, let us consider $A(G_1)$.

- *Case 1:* $A(G_1) \in T_0$ $L(\ell_s, A(\hat{G}), G_0) = 2$ but $L(\ell_s, UNI(G_0), G_0) = 2$. Therefore, $B(A, G_0) = 0$.

- *Case 2:* $A(G_1) \in T_1$ This requires that there be at least one connection from $A$ to $B$.

  Let $m(d)$ be the maximum number of nodes that can exist in an subtree rooted at a depth of d such that all nodes satisfy the individual rationality property.

In $G_1$, assume that $A$ and $B$ select $z = m(1)$. Further, let us assume that $u(.)$ is such that at depth 2, the subtree can support $m(2)' < m(1)$ nodes.

In the unicast tree, there is no traffic between A and B, or $L(\ell_b, UNI, G_1) = 0$. Therefore if $A$ is to have non-zero benefit we must have $L(\ell_b, A(G_1), G_1) \leq 1$. Since we must serve the users at B we must have at least one connection from $A$ to $B$. Thus, we have $L(\ell_b, A(G_1), G_1) = 1$.

Consider the single connection from $A$ to $B$: $(A_i, B_j)$ where $A_i$ and $B_j$ are nodes. This means that $B_j$ is at depth 2. However, since there is exactly one link into $B$, all $M - 1$ other nodes must be descendants of $B_j$. Therefore, we have a tree of size $z = m(1)$ rooted at depth 2, which violates the definition of $m(.)$.

Consequently, $A(G_1)$ does not satisfy the IR condition and thus it is not the case that $B(A(G_1), G_1) = 0$.

□

## 5.5 A Repeated Game Model

The negative results of the prior section are troubling for several reasons. They do not bode well for such overlay systems. More relevant however is that they do not precisely correspond with our observation of reality. Indeed, on a limited scale, such applications can be successful. In fact, some have recently argued *against* worrying about incentive mechanisms for newer systems precisely for this reason [48]. A more complete model for this problem *must* explain this phenomenon of cooperation but also *should* do so in a fashion that aids the design of systems to engender such cooperation, if desirable.

Our core observation is that while selfish users want to maximize their utility at any point in time, in most systems **even selfish users want the system to exist in the future.** If a multicast topology is being considered for an application, we assume that it is required (to some degree). Such motivations could be a source trying to avoid a large bandwidth bill or an end-user trying to avoid being shut down by an ISP or IT group. In such scenarios, no user cares about the state of the network per se *but* all users want the overlay to continue to exist and thus indirectly care about the network's health. This is a dynamic that is well captured by a repeated game model.

While our model is heavily inspired by the repeated game models of the game theory literature, it is stark contrast with the existing Computer Science literature on this class of problems. For example, Mathy *et al*[67] model cheating in multicast overlay trees by randomly assigning a given percentage of users to be cheaters. These cheating nodes always cheat and the other nodes are not selfish and always faithful. This has the weakness that the desire to cheat is independent of the user's experience and thus exogenous to the model. Similarly, Feldman *et al* look at the related problem of engendering cooperation in the context of peer-to-peer file sharing. Here they develop threshold [37] and randomized (mixed) [36] strategies which produce a non-trivial equilibrium. While the notion of repetition and cooperate is implicit in their models, the relevant parameters (the threshold or mixing proportion) is again exogenous to the model. By contrast, *in our model, the motivation for cooperation is endogenous to the model.* All users are selfish and the propensity to cheat is based not on some external parameter. Rather the user's actions are based on the experience of the particular user and the degree to which she can improve her utility by cheating. As we will see in Section 5.7.2, **this difference also enables our model to be of use in designing systems where the propensity of users to deviate is diminished.**

### 5.5.1 Key Modeling Decisions

Before presenting the formal model, we examine two key modeling decisions. First, our model assumes that the motivation for cooperation is that if too many users cheat, network load will increase and the overlay will be shut down. In practice, there may be other or alternative reasons for cooperation, which we present below. Also, for nodes to remain faithful, they must have some way to appreciate the state of the network and how their actions impact this state. Another assumption our model makes is to rely solely on repetition as the incentive for faithfulness, and not consider such alternative concepts as epsilon-equilibria. Therefore, before presenting the formal model, we discuss these assumptions and why they were made.

#### Goals of a Multicast Network

Recall that the model of cooperation here is that users' desires to cheat are mitigated by their concern for the future. To develop a model and simulator, we must formalize this concern. We do this by creating a metric which corresponds to the health of the network.

For non-trivial repeated outcomes to result, the metric must satisfy five key properties. (1) The metric must be sufficiently bad for unicast topologies but sufficiently good given faithful users and an overlay multicast topology. (This is the core assumption of overlay trees.) Further it must be the case that (2) this metric degrades as users cheat and, if the metric is sufficiently bad, the network will cease to exist in the future. The users must have some (3) signal of the metric, and (4) means of appreciating the impact that their cheating has on this metric. Finally, (5) users must have some understanding of how altering the metric affects the chances of the network ending. Note that (3), (4), and (5) can all be delayed and/or noisy signals.

One motivation which satisfies these requirements is network load. For the remainder of this chapter, we define network load as the number of unique packet-hops, which is analogous to average stress. If the load is too high, or close enough to the unicast load, the network could be shut down by third party such as an ISP or IT organization. This clearly satisfies properties (1) and (2). In practice, users can obtain a reasonable indication of the state of the network (3) from the source and/or from traversing the network itself. Similarly, the source could provide information on the response function (5). The source's information could a data-feed or a more crude method such an email to users regarding the pressure the source is getting from its ISP. To address their impact (4), users need only understand the load of the stream and the distance to the source relative to the distance to their parent, all of which can be observed directly. Finally, we note that in practice, the users do not even need a noisy signal of the entire response function (5), but only the local derivative. To capture these factors, in our model, we assume that the users have perfect information as to the state of the network and the direct impact of their actions. We then relax these assumptions in Section 5.8. We also assume that users have no knowledge of the impact their actions have on other users' willingness to cheat. (We discuss this assumption more formally in Section 5.6.9.)

Another motivation for a multicast tree is source load. Lowering the load on the source could provide scalability or perhaps decrease the operational costs, making the system possible. This satisfies properties (1) and (2). Further, the dynamics and availability of information here very similar to overall network load metric above. In this case the information is even more readily available to the source. As such, it can satisfy (3), (4), and (5).

A third motivation for a multicast tree is increased quality given fixed resources. With only the unicast topology, the load on the network, users, and/or source causes degradation in the quality of the experience. This satisfies property (1). Given such a

scenario, nodes are incentivized to reduce network stress and load on their ancestors in the tree so as to ensure high quality. Here decreased quality could cause nodes to leave the system. As users leave the system, the motivation for the source to continue broadcasting is likely diminished – for example, ad revenue may decrease or the effort to produce the content may no longer be worthwhile. This satisfies property (2). Last, while the observations may be more noisy, the users can satisfy properties (3), (4), and (5) in similar fashion to the above examples.

Given that all three motivations (network load, source load, and quality) satisfy these five parameters in similar fashions, we focus on one – network load – for the remainder of this chapter. This is a motivation faced by many end-user broadcast systems, lest lest the ISP or relevant network operators take action against the network load. This metric is also very similar to the metric of average stress found elsewhere in the literature. We believe that this assumption does not impact our conclusions and does not qualitatively affect our results.

#### Alternative Factors

Another major design decision is to not involve additional tools which help to explain cooperation such as epsilon-equilibria or generosity factors. The epsilon-equilibria notion captures the fact that that there is some effort associated with cheating and has been suggested as a good model for networking problems [13]. However, the cost of cheating could be as simple as downloading and installing an appropriate binary, which is quite easy. Given that a binary for the faithful application itself must be installed, cheating may require no additional work. Alternatively we could incorporate some of the proposed ideas regarding altruism factors. However, we feel that the repeated dynamic – not the barriers to cheating (epsilon equilibria) or exogenous generosity – is the best explanation for why we may see cooperation in these networks. While the approaches could be complimentary, we do not include them here for clarity.

### 5.5.2 Formalizing the Repeated Model

Based on these assumptions, we can now model the entire game. First, we restate the baseline definition from Section 5.4.2 with some elaborations, then we define the network load metric and the ISP response function. We then present the equilibrium equations.

### Instance Definition

- A network, $G = (V, E)$, $V$ and $E$ are finite.
- A set of nodes N to be served, $N \subset V$
- A single source, $s \notin N$, $s \in V$ which sends a stream of data to all nodes directly connected to it. The source has infinite capacity and constant utility.
- A single, atomic, piece of content to be sent from the source to the end-users.

### Tree Related Definitions

Given a problem, we have an algorithm, $A$, which constructs an overlay topology, $T$.

**Definition 6.** *The* load *of an overlay topology, denoted $L(T, G)$ or $L(T)$ is defined as the sum of the load on the individual links, or:*

$$L(T, G) = \sum_{\ell \in E} L(\ell, T, G)$$

*where, as before, $G = (V, E)$ and $L(\ell, T, G)$ is the load on link $\ell$ in network $G$ given the overlay topology $T$.*

**Definition 7.** *A* valid *tree is one that satisfies the requirements outlined in Section 5.4.2:*

1. *All $N$ nodes are connected by $T = A(G)$; and*
2. *All nodes satisfy the individual rationality constraint, that is, $u_i(d_i, c_i) \geq 0 \ \forall i$.*

To satisfy properties (1) and (2) from Section 5.5.1, the ISP response for a given load $L$ is defined as a function of the load of the unicast tree and the load of the faithfully constructed overlay tree. More formally we define $T_{max}$ to be the unicast tree. That is, $T_{max} = (s, i) \forall i \in N$. $T_F$ is defined to be the faithfully constructed multicast overlay tree, adjusted by connecting nodes who are further from their parent than the root to the root. Let $L_F = L(T_F)$ and $L_{max} = L(T_{max})$.

A response function represents the probability that the game will end in a given period. Therefore, to correspond to our goals, a valid response function $R(L, L_F, L_{max})$, denoted $R(L)$ must satisfy three properties:

1. $R(L_F) = 1.0$ (The network will exist if users are faithful. [6])

[6] We could allow for $R(L_F) < 1.0$, but chose not to for clarity.

2. $R(L_{max}) < 1.0$ and $R(L_{max}) \geq 0$ (The unicast network will continue to exist with some probability strictly less than 1.0.)

3. $\frac{R(L)}{dL} \geq 0$ for $L_F \leq L \leq L_{max}$ (For valid $L$, $R(L)$ increases with $L$.)

### User Model

Given an overlay tree, $T$, we have:

- $d_i(T)$ denotes the depth of node $i$ in tree $T$ and $c_i(T)$ denotes the number of children that $i$ supports in $T$.

- $u_i(T) = u_i(d_i(T), c_i(T))$. $u(.)$ is strictly decreasing in both $d_i$ and $c_i$.

As discussed in Section 5.5.1, users have knowledge of $L(T)$ and $R(.)$, but not $T$ nor the history of plays that produced $T$.

In Section 5.8.2, we consider a variety of potential response functions. Two such functions are plotted in Figure 5-15.

### User Decision

At any point in time, a user may accept the faithful position in the tree $(d, c)$ with associated load $L$ or cheat and obtain some alternative $(d', c')$ with new load $L'$. By construction we have $d' < d$ and/or $c' < c$. Thus, we have $u(d', c') \geq u(d, c)$ but such a deviation causes the new load to be greater, that is $L' > L$. As discussed above, while we later relax these assumptions in simulation, we assume that all users have perfect knowledge of $R(.)$, $L$, and $L' - L$. We also assume that in practice they can observe $d$ and $c$.

Devoid of any monitoring or enforcement mechanism, if a user chooses to accept $(d, c)$ (other than $(1, 0)$), it must be the case that the decreased chances of the network's future existence outweigh the benefits of changing the position in the tree. We can model this formally. In doing so, we make a key assumption that users have a limited understanding of the impact their actions will have on other users. In particular, each user moves as if her move will be the last ever. In practice, when one user cheats, this may precipitate more cheating by others. However, for a user, reasoning about this process is not simple. Therefore, our model assumes limited rationality – users have no ability to reason about this impact and do not consider it. The effect of this is that users are more likely to cheat than appropriate – making our results of cooperation conservative.

Taking these factors, we have that for a user $i$ not to cheat:

$$\sum_{t=0}^{\infty} \delta^t R(L)^t u_i(d,c) \geq u_i(d',c') + \sum_{t=1}^{\infty} \delta^t R(L')^t u_i(d',c') \tag{5.1}$$

The left-hand term represents the discounted payoffs to cooperating. The first right-hand term represents the one-stage payoff to cheating and the last term represents the continuation payoff from that cheat.

Simplifying and putting the equation into standard form:

$$\frac{u_i(d,c)}{1-\delta R(L)} \geq u_i(d',c') + \delta R(L') u_i(d',c') * \frac{1}{1-\delta R(L')} \tag{5.2}$$

$$u_i(d,c) \geq (1-\delta R(L)) u_i(d',c') + \delta R(L') u_i(d',c') * \frac{1-\delta R(L)}{1-\delta R(L')} \tag{5.3}$$

Similar to our analysis in Chapter 4, equation (5.3) equation allows examination of how the various parameters may impact a user's desire to cheat. Informally:

- *Patience of Users* As $\delta \rightarrow 0$, the right hand side goes to $u_i(d',c')$. Since $u_i(d',c') > u_i(d,c)$, we see that users become more likely to cheat.
- *Benefit to Cheating* The benefit to cheating is $(u(d',c') - u(d,c))$. As this increases, the right-hand side becomes large relative to the left, meaning that users are more likely to cheat.
- *Cost to Cheating* By contrast, as the cost of cheating $R(L') - R(L)$ becomes large, the final term becomes small, making users less likely to cheat.

## 5.6 The Simulator Methodology

This problem of multicast overlays is complex with many variable attributes including the number of players, utility functions, overlay tree topology, and underlying network topology. As a result of this complexity, it is not possible to theoretically analyze the behavior of the entire system. Therefore, we use simulation to better understand the system dynamics and to gain intuition for design.

In this section, we describe a simulator designed to capture and vary these parameters. First, we overview the simulator. We then discuss several key assumptions implicit and explicit in the simulator design.

### 5.6.1 Overview

The goal of the simulator is to model the decisions, actions, and interactions of self-interested user-nodes who are part of a single overlay topology.

1. The simulator takes a set of inputs:
   - A synthetic network topology.
   - A user utility function ($u(d, c)$)
   - A number of nodes ($N$)
   - A discount factor ($\delta$)
   - A tree formation algorithm
2. The simulator randomly selects a source and $N$ nodes in the topology to be end-users and constructs the overlay tree using the specified algorithm.
3. Each node learns its depth and number of children and receives a signal of the efficiency of the overlay system.
4. The simulator considers each node in the topology and allows it to take action. The permissible actions are: 1) connect to root and 2) drop child.
5. Step 4 is repeated until no node wishes to act.
6. Statistics are collected and reported.

### 5.6.2 Implementation of NICE

When the tree formation algorithm used is NICE, the simulator uses a custom-built implementation of the NICE protocol. The NICE modules takes as input a network topology, a set of $N$ nodes, a source, and a cluster size parameter ($k$). The simulator then builds the the NICE tree, bottom up, using a distance metric of hop count, which corresponds to our definition of load (as discussed in Section 5.5). As nodes enter and leave the tree, the NICE module performs the appropriate cluster joins and splits as specified by the protocol. Furthermore, to simulate the limit of the constant refinement process, upon any change to the tree, the tree is rebuilt from the bottom (layer 0) of each rooted subtree.

One necessary modification of the NICE protocol is that we allow for multiple top-layer nodes in steady-state. This is done for multiple reasons. In order that no

node has negative utility, the depth of the tree and the number of children of any node must be bounded. Yet, the tree protocol must always allow for additional children of the source as the number of nodes increases. Furthermore, if the NICE protocol did not allow for this, a pathological behavior could result where nodes kept increasing their claimed layer (though fabricated nodes) to claim leadership of the top level, thereby creating an instability in the tree. Finally, note that multiple source children *are* allowed in the standard NICE protocol in cases such as partitioned networks. Therefore, it is possible to for cheating nodes to produce such a topology if they need to.

### 5.6.3 Implementation of Cheating

The simulator allows for two potential cheating actions: connecting to the source and dropping a child. As discussed, there are a variety of ways to cheat and a variety of partial defenses that could be used. Our simulator's implementation of the cheats is designed to be simple but representative.

Nodes are free to connect directly to the source whenever they want. When a nodes does so, it maintains the structure beneath itself. As such, this essentially creates another NICE tree rooted at the same source. This tree continues to follow the NICE protocol.

Dropping a child is a slightly more subtle operation due to the complexity in reconfiguring a tree. While there are multiple subtle ways to drop a child, the simplest is to simply stop sending the child any data. In NICE, the orphaned node would attempt to perform a new leader election within its cluster and connect to the higher layer. However, this higher layer could also thwart the orphan. While it is conceivable that we could detect when the leader at the higher layer would also reject it, this procedure of determining the exact reconnection point can be quite complex. Therefore, for simplicity, when a child is orphaned, the simulator reconnects it directly to the source. Further, after dropping a child, the parent creates an artificial child to avoid receiving a new child. Given this behavior, a parent when considering whether or not to drop a child, first considers the child who would generate the least amount of additional load at the source. This maximizes the expected system lifetime, and hence, the future payoff of the parent.

### 5.6.4 Randomization

For a given experiment (which defines $N$, $\delta$, and $u(.)$), each run is defined by a network topology, a source node, and a set of user nodes. The data presented varies the inputs in two dimensions:

- Network Topology: As discussed in Section 5.6.7, we use three different network topologies. Each is randomly generated by BRITE, with the same parameters.
- Random Seed: Each run is defined by a random seed. The seed selects the source and the users from the network topology. It also defines the order via which the simulator will iterate through the users.

Unless otherwise noted, each experiment uses 30 random seeds and 3 topologies for 90 runs per data point. To facilitate comparison, we use the same topologies and seeds across all of our experiments.

Most graphs present only the mean value over the trials. However, the individual trials of all graphs are qualitatively similar. Therefore, we feel that the mean is a sufficient metric, unless otherwise noted.

### 5.6.5 Utility Function

The assumptions about QoS and the incentive for cheating is embodied in the utility function. We use utility functions of the following form:

$$u(d, c) = \gamma - \lambda\sqrt{d} - \beta c$$

where $\gamma$, $\lambda$, and $\beta$ are all parameters configurable in the simulator. The selection of the square root function for the depth is designed to capture the decreasing marginal disutility of increasing depth. The linear function over the number of children is based on the fact that some of the costs of supporting children (e.g., CPU, bandwidth) scale linearly with load. Finally, we have the constant $\gamma$ term to capture the fact that while depth and children do matter, another important component of the utility function is simply obtaining the content. Unless otherwise noted, the simulations in this paper use $\gamma = 10$, $\lambda = 1.0$, and $\beta = 0.25$; in Section 5.8.1, we discuss a range of alternate utility functions.

### 5.6.6 Response Function

The response function of the ISP plays an important role in the behavior of the system. The simulator does not model the actions of the ISP per se, but rather uses the response function when evaluating the utility of different actions. Unless otherwise noted, the simulator uses the following linear response function:

$$R(T) = \frac{L(UNI) - L(T)}{L(UNI) - L(F)}$$

where $L(.)$ is the load operator, UNI is the unicast tree, and F is the tree produced by faithful users. The function is graphed in Figure 5-4.

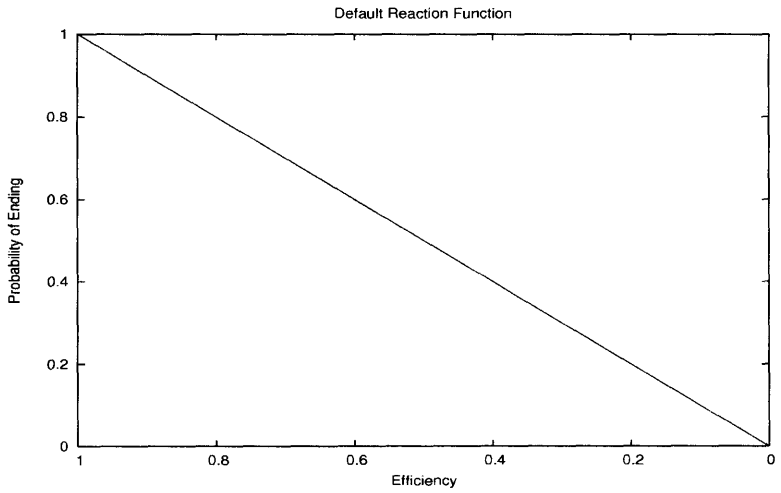
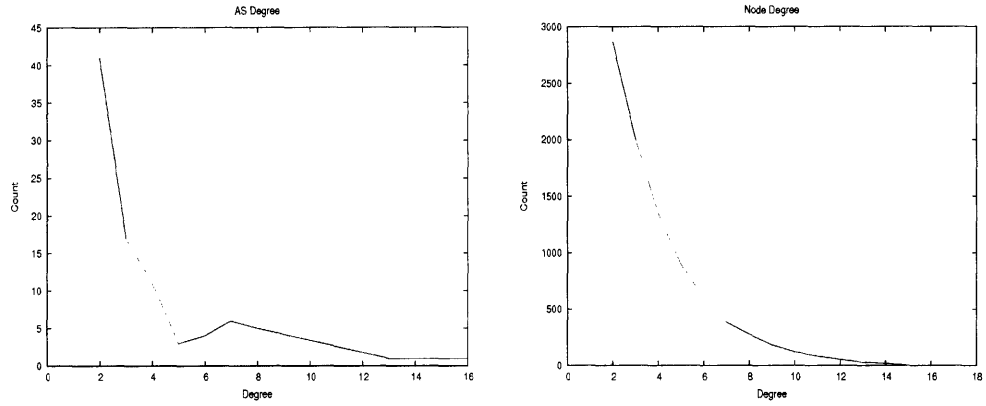



Figure 5-4: The Reaction Function. As efficiency decreases, so too does the chance that the network will end.

Section 5.8.2 presents an analysis of a wider space of response functions, demonstrating the relative insensitivity of the conclusions to the choice of functions.

### 5.6.7 Network Topologies

To generate the underlying network topologies, the simulator uses the Boston university Representative Internet Topology gEnerator (BRITE) [68] . BRITE uses heavy-tailed models to produce inter-AS and intra-AS models. In our simulations we use the BRITE Barabasi model, a preferential attachment model, for the inter-AS connectivity. For intra-AS (router) connectivity, we use the BRITE Waxman model. All models have 3000 nodes.




(a) AS Degree (b) Router Degree

Figure 5-5: Degrees of ASes and Routers in the Synthetic Network

While BRITE is a good simulator, it is not perfect. Indeed, synthetic topology generation is an active topic of research and there is good reason to believe that current topology generators, such as BRITE and GT-ITM [69] are not yet fully representative of true topologies. One particular concern is that while heavy-tailed distributions may be appropriate for AS connectivity, routers with massive numbers of connections (as is dictated by heavy-tailed distributions) may not be practical [63]. Figures 5-5(a) and 5-5(b) plot the degree distribution for ASes and nodes respectively cumulative over the sample topologies used. In particular, Figure 5-5(b) shows that the maximum degree of any node is 18 and only 10 nodes have degree 15 or more, which represents only 0.5% of nodes in a given topology. Therefore, we conclude that while the topologies may not be perfect, this issue is not of concern for our analysis.

### 5.6.8 Alternative Design Decisions

The simulator, as presented, embodies a few key design decisions and assumptions regarding the approach to finding trees and user behavior within the trees. However, we feel that the conclusions drawn are qualitatively similar to those which would have been obtained from alternative sound assumptions.

One implicit decision of the simulator is to construct trees instead of searching the space of potential trees. For many formation protocols, various faithful trees could be constructed depending on ordering. Once built, the ordering of the cheating could also cause different stable topologies result. Therefore, an alternative approach is to construct all possible topologies and evaluate which are stable. This has two

downsides in that it is computationally difficult and also provides little justification for why one topology may arise versus another.

Another key decision is whether or not actions, once taken, are permanent. In practice, when a node cheats, for example, inventing children to move up the tree, this cheat need not be permanent. After some period of time, based on the actions other nodes, the node could choose to become faithful. However, such a strategy would likely be quite complicated for a node to implement effectively. Instead, our model takes the approach that cheating is a permanent decision, though (as described above) after cheating the user may still choose to faithfully implement NICE in her new subtree.

### 5.6.9 Proof of Equilibrium Results (Optional)

For simplicity, the strategies used by the agents here do not anticipate the reactions of other players. Therefore, there may be additional, more efficient, equilibria which are possible. In our analysis, we examine properties of the stables points of the simulator – when no user wants to chat. Under reasonable assumptions, these stable points are equilibrium outcomes even if players had a perfect ability to understand future reactions.

While a formal proof follows, the institution is relatively simple. Consider an overlay topology $T^*$ that is stable in the simulator. That is, no player wishes to either root itself or drop a child considering only that action in isolation. If $T^*$ were not an equilibrium outcome (given perfect foresight), then a player must be able to make a beneficial move, considering the reactions of the other players. Since the only difference between these two cases is the other players' reactions, this requires a player $i$ to be able to move in a way that precipitates a move by another player, $j$, which is beneficial to $i$. We can show that under reasonable assumptions, this is impossible.

In this paper we consider two tree formation protocols, NICE, which has been already presented, and Naive Min Cost, which we present in Section 5.7.2. One salient feature of NICE is that while most changes to not affect the tree, some changes, over time can produce intricate reorganizations. Thus, in rare cases, it may be possible for a node to move which causes the protocol to further rearrange the tree in a way that benefits the node. We call such events *beneficial induced restructuring events.* In practice, not only are such events relatively rare (since it involves at least two cheating moves leading to a positive outcome) but it would be very difficult for a node to properly foresee such a move. Further, the move will likely need to involve

nodes not nearby in the tree as simple moves (like disconnecting from the parent or having a child leave) can be accomplished directly by the node. We therefore feel the following assumption is very reasonable:

**Restructuring Assumption**: We assume that no node can cause a beneficial induced restructuring event. That is, there is no action for any player $i$ that initially reduces her total payoff, but induces the tree-formation protocol to restructure the tree, which in turn induces some node $j$ to cheat, and finally results in a topology where the discounted payoff to $i$ is greater than in the original topology.

Again, this assumption is only meaningful for protocols like NICE which attempt to re-balance the tree. For protocols which produce a static tree or for architectures in which there is no tree formation protocol, this assumption is not required. Also, note that this assumption is not used at all in the simulation, but rather, is needed in order to interpret the simulation outcome as an equilibrium. We can now use the restructuring assumption to formalize the preceding argument into the following theorem.

**Definition 8.** *Let $\alpha$ be the strategy represented by the logic of the simulator. That is, $\alpha$ prescribes that $i$ moves from $(d, c)$ to $(d', c')$ iff:*

$$u_i(d, c) \geq (1 - \delta R(L))u_i(d', c') + \delta R(L')u_i(d', c') * \frac{1 - \delta R(L)}{1 - \delta R(L')}$$

*Further, per the implementation of the simulator, $\alpha$ first considers rooting itself and then considers child drops, in order from the least rooting cost to the most.*

**Theorem 12.** *Let $T^*$ represent a stable point for players in the simulator. Assume that the restructuring assumption holds. Then, starting from $T^*$, always playing "Stay" is a Subgame Perfect Equilibrium.*

*Proof.* The action space available to each node is { Drop Child $j$, Root Self, Stay }. Assume that the strategy of all players always playing "Stay" is not a subgame-perfect equilibrium. Then, there must be player $i$ who can improve her payoff by deviating from this strategy at some point in time. Consider the first player $i$ who can profit from a deviation; it follows that $i$ can profit from the deviation in the first round after reaching topology $T^*$, because "Stay" plays do not alter the state of the game. Let us call the deviating action – which must be "Root Self" or "Drop Child" – $D$

Let $x$ be the total payoff to node $i$ from playing "Stay", and let $x'$ be the total payoff to node $i$ from playing $D$, *assuming that all other nodes continue to play*

*"Stay"*. But this is exactly what the simulator examines and by definition it already considered and rejected $D$. Thus, if $T^*$ is stable in the simulator, it must be the case that $x \geq x'$.

Now let us consider the impact of paying $D$ on other nodes. Let $x''$ be the payoff to $i$, taking into account the impact of playing D on other nodes. Using the restructuring assumption, this payoff cannot be greater than $x$– there are no indirect benefits to $i$. Thus, we must have $x \geq x''$ – contradicting the assumption that $D$ was a profitable deviation. □

## 5.7 Core Results

In this section, we present the three core results from the simulator and discuss their significance for system design. Our three core results are:

1. System efficiency decreases with $\delta$.
2. System efficiency decreases with $N$.
3. For the NICE protocol, under reasonable assumptions system efficiency for sufficiently impatient users increases with $k$, the cluster size.

The first two results are fundamental to the thesis of this paper and validate our basic assumptions and model – that concerns about future existence can mitigate cheating in the absence of external enforcement mechanisms. This is in line with standard repeated game results. The third result is novel insight gained specifically from this problem. In short, we find that the NICE protocol naturally exploits the tradeoff between depth and number of children by assigning nodes at higher levels more children. This means that with even mild amounts of selfishness, NICE trees can outperform a centralized min-cost spanning tree protocol. Further, since this tradeoff is already an explicit parameter in the protocol (cluster size), it can be adjusted in various ways.

To provide insight and intuition, the results presented in this section consider a simple model. In particular, we consider the basic utility function (from Section 5.6.5), no noise, and linear reaction curve. In Section 5.8, we relax these assumptions to show that the main conclusions are robust to a wide range of operating environments consistent with our base assumptions.

Many of the graphs in this section plot the impact of a particular parameter on system efficiency. Efficiency in this context is defined to be the fraction of the

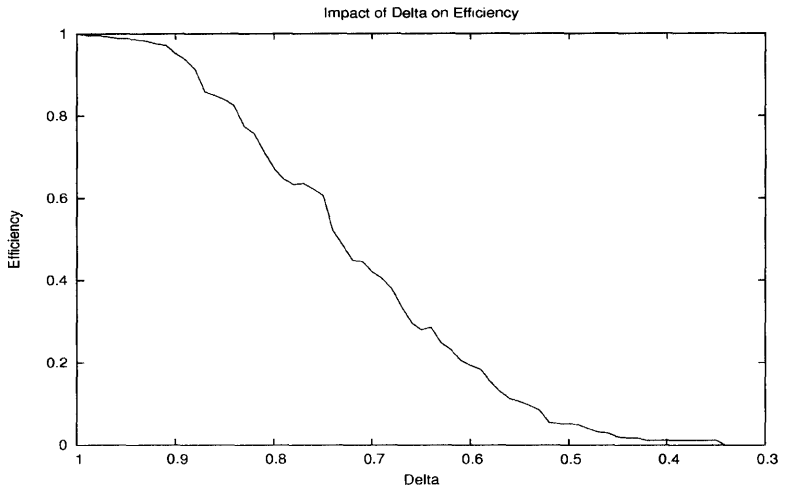



Figure 5-6: Efficiency versus Delta. (Note that $\delta = 0$ corresponds to the one-shot game.)

improvement provided by the overlay that is realized in the face of selfish users. Formally, we define it as:

**Definition 9.** *The efficiency of a particular tree, $T$, is defined as*

$$\frac{L(U) - L(T)}{L(U) - L(F)}$$

*where $L(.)$ calculates the network load of a tree, $U$ is the unicast tree, and $F$ is the overlay tree for the same network and users, with all users being faithful.*

### 5.7.1 Cooperation in Practice

As expected, system efficiency decreases with delta. This can be seen simply in Figure 5-6, which examines a topology of 50 nodes.[7] The steep slope as $\delta \rightarrow 1.0$ results from the $\frac{1}{\delta}$ terms in Equation (5.3 ). It is also useful to note that $\delta = 0$ corresponds to the negative result from the one-shot game presented in Section 5.4.

Our second core result is that efficiency decreases with the number of nodes. This is important to understand as it is an important dimension to consider for scalability concerns. Figure 5-7 builds on Figure 5-6 by adding $N = 10$ and $N = 100$. Increasing $N$ has two impacts on efficiency – it increases the minimum delta at which cheating occurs and increases the rate at which efficiency approaches zero.

[7] As discussed previously, each data point represents the average of 90 separate randomly generated trials.

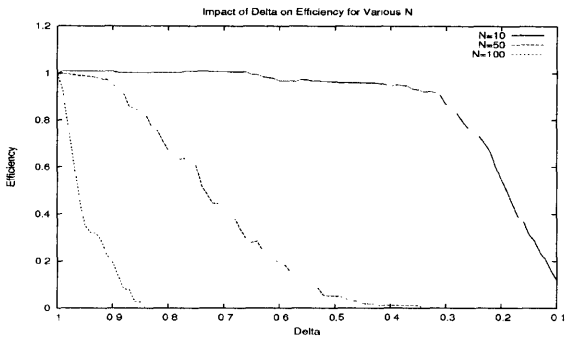



Figure 5-7: Efficiency versus Delta for N $\in$ {10, 50, 100}.

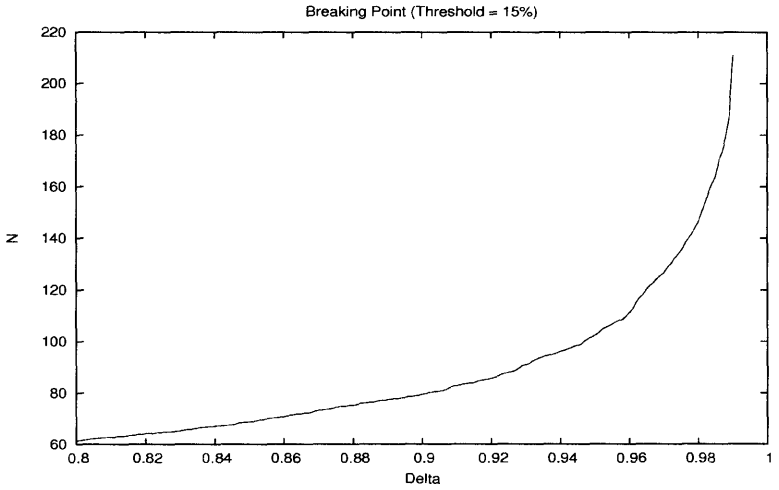



Figure 5-8: Breaking Point as a Function of $\delta$. The breaking point is the minimum value of $N$ such that efficiency falls below 15%

An alternative way to view the impact of the number of users is to examine the maximum value of delta such that efficiency falls below a given threshold. We define this critical point to be the "breaking point." Figure 5-8 plots the breaking point as a function of delta with a threshold of 15%. For example the graph contains the point (0.94, 100). This means that with $\delta = 0.94$ the maximum number of nodes in a network such that the efficiency is at least 15% is 100. As would be expected, the curve is very steep as $\delta \rightarrow 1.0$.

### 5.7.2 Robustness of Tree Formation Protocols

The first two core results beg the question *"How does the shape of tree impact its robustness given impatient users?"* Of course different tree formation algorithms and

protocols can form trees with vastly different structure. Thus, instead of heavyweight mechanisms (e.g., payment schemes or monitoring) an alternative approach to building more robust trees is to alter the protocol or algorithm appropriately.

### NICE vs Naive Minimum Cost Spanning Tree

To begin our analysis, we compare NICE to a centralized algorithm which creates the minimum cost spanning tree. As in [83], this centralized algorithm may be run at the source and with faithful nodes, will minimize the cost metric, network load. Since selfish agents can perturb the tree, we instead call this algorithm Naive Min-Cost Spanning Tree (NMC).

There are however several ways in which a node can cheat this centralized algorithm, depending on the particulars of the algorithm. If the nodes simply report their distances, as in [83], then a cheating node can deflate its distance to the root and inflate its distances to its neighbors [67]. Alternatively, if the algorithm limits the number of children assigned to a node, the basic NAT-based Sybil attack will enable the node to have no children. Further, if the algorithm attempts to minimize or otherwise bound the depth of the tree (to reduce stretch or to preserve quality), the NAT attack will enable a node to connect directly to the root. For the purposes of this analysis, we assume that one or more of these techniques will permit the action space that we simulate for NICE – a node can drop any children and/or connect to the root if desired.

Not surprisingly, for $\delta = 1.0$, NMC outperforms NICE, as can be seen in Figure 5-9. The fact that NICE performs worse is based on the fact that NICE makes myopic decisions based on only local information. However, note that as in [6], NICE still performs relatively well.

However, we see that NICE is far more robust, and thus far more efficient, when faced with selfish users. Figure 5-9 plots the load on the network for both NMC and NICE relative to the load of the minimum cost tree, varying delta.[8] While the NMC algorithm outperforms NICE for $\delta = 1.0$, its efficiency rapidly decreases. NICE is therefore able to perform better for a wide range of $\delta$.

Careful inspection of the tree structure lends insight into why NICE is robust but NMC is fragile to selfishly minded users. With the NICE trees, a cluster leader at depth $k$ is by definition a cluster leader at all depths $k' > k$. As such, the benefits to being higher in the tree are mitigated by supporting more children. In contrast, the

[8]Because we are comparing two different topologies, we plot load on the y-axis.

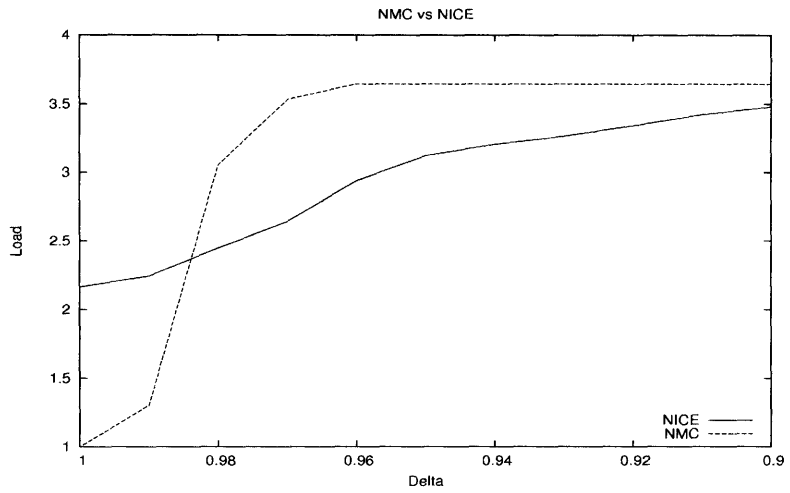



Figure 5-9: Performance of Naive Min-Cost vs NICE. While the NMC is superior for faithful users, with even modest discounting NICE performs better.

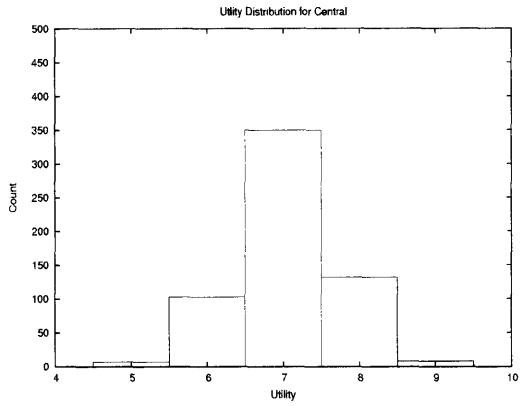
minimum cost tree has no such guaranteed structure. While of course it is possible by sheer luck to have a robust topology, this is unlikely. We can further appreciate this relationship by looking at the distribution of utilities in the faithful trees. Figure 5-10 the per node utilities (bucketed to the integer value) of the node at tree-creation time. Here we see that the disparity in user utilities is greater for the NICE algorithm ($\mu$ = 8.37, $\sigma$ = 0.54) than for the Naive Min-Cost Algorithm ($\mu$ = 7.46, $\sigma$ = 0.61).

### Applying the Lessons to NICE

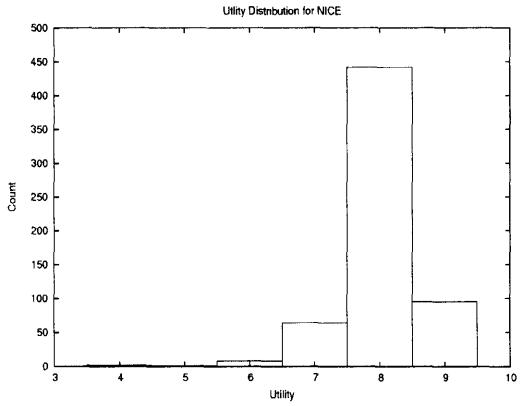
This intuition can be used to increase the robustness of NICE itself. A structured tree where nodes near the top are asked to bear an appropriate fraction of the burden is more robust against user incentives. This suggests that just as we gained robustness moving from NMC to NICE, we can, in some cases, also gain by increasing the cluster size.

Recall from Section 5.2 that the cluster size of the NICE algorithm defines the tree structure and the efficiency of the tree when users are faithful. Figure 5-11 depicts this relationship. In general, we find that low k, 2 or 3, minimizes load. (While [6] and [67] do not present an analysis of load, they do use $k = 3$.) Obviously for $k > N$, we essentially have a unicast tree with the source sending the stream to one node who then sends it to everyone else. For $N > k > 3$ efficiency degrades.

However, Figure 5-12 shows how load scales as we decrease delta for various $k$ and the NMC algorithm. A load of 1.0 represents the load of NMC at $\delta = 1.0$. The




(a) Distribution of Utility for the Naive Min-Cost Algorithm ($\mu = 7.46$, $\sigma = 0.61$)




(b) Distribution of Utility for the NICE Algorithm ($\mu = 8.37$, $\sigma = 0.54$)

Figure 5-10: Utility Distribution for NMC and NICE at $\delta = 1.0$. NICE induces homogeneity across positions in the tree.

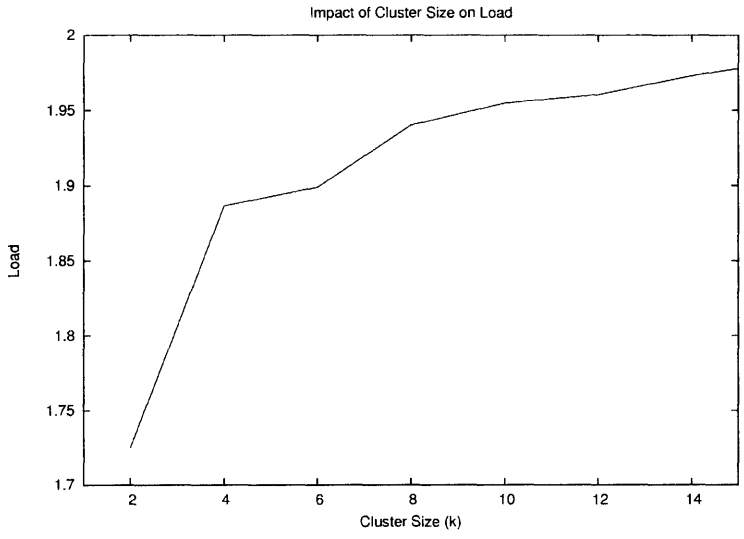



Figure 5-11: Load of NICE versus Cluster Size (k) at $\delta = 1.0$ (normalized to the load of NMC at $\delta = 1.0$.)

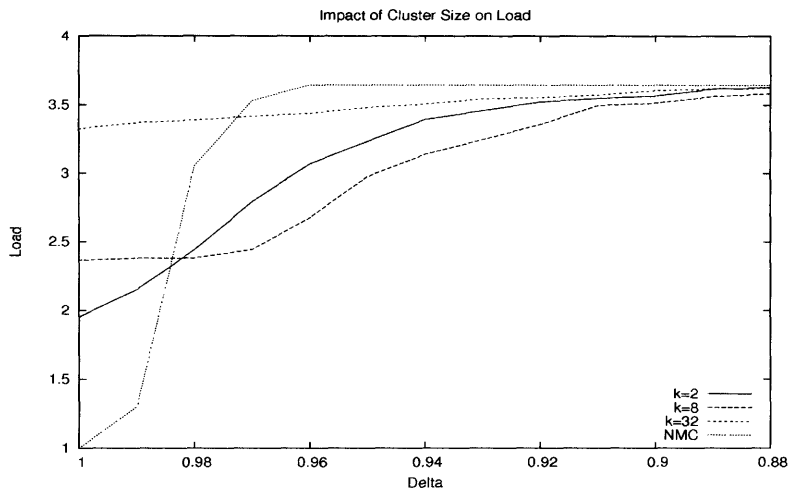



Figure 5-12: Load of NICE $k \in \{2, 8, 32\}$ and NMC versus $\delta$ (normalized to the load of NMC at $\delta = 1.0$.)

topologies with larger $k$ perform worse for faithful users but degrade significantly less as $\delta \to 0$. This suggests that over a range of $\delta$ values, while increasing $k$ may decrease efficiency for faithful users, it can be used to increase the robustness – and thus the realized efficiency – of the system.

## 5.8 Insensitivity of Results (Optional)

The previous section presented the core results and intuition for a single and relatively simple model. In this section, we examine the impact of several important assumptions. For each, we show that while they can influence the quantitative results, the key qualitative results are unchanged.

### 5.8.1 Insensitivity of Conclusions to Utility Functions

Recall that the utility function used in the chapter thus far is:

$$u(d, c) = \gamma - \lambda\sqrt{d} - \beta c$$

with $(\gamma, \lambda, \beta) = (10, 1.0, 0.25)$. This function captures our assumptions regarding the importance and direction of depth and number of children on the user's experience. However, the intuition and conclusions presented extend to a large class of utility functions that correspond to our assumptions.

Figure 5-13 plots efficiency versus delta for a variety of parameter settings for the utility function. (Each utility function is represented in the legend by a ($\gamma$, $\lambda$, $\beta$) tuple.) As the parameter values increase, the rate at which efficiency diminishes also increases. This is as expected as higher parameters correspond to a greater incentive to cheat, and vice versa. Despite these differences, the high level conclusion that non-trivial equilibria can exist for sufficiently large $\delta$ can exist in our model remains.

Figure 5-14 examines the impact of the utility function on relationship between $k$ and efficiency as a function off $\delta$. [9] In the previous section we saw that larger $k$ could increase efficiency with selfish users. Again, as a baseline, we see that across all graphs at $\delta = 1.0$ the smaller values of $k$ perform better. However, as $\delta$ increases, this performance difference can significantly diminish, and in some cases become inverted – as seen in the example of the last section. Further, examining Figures 5-14(a) and 5-14(b) which correspond to the range of utility functions most in line with our assumptions, $k = 8$ is the most efficient of the 3 curves plotted for a large range of $\delta$.

We also consider the extreme end of the parameter spectrum with $\beta = 0.5$ and $\beta = 0.75$. For perspective, moving from a depth of $d = 2$ to $d = 1$ increases utility by $\sqrt{2} - \sqrt{1} \approx 0.41$ and from $d = 3$ to $d = 2$ is $\sqrt{3} - \sqrt{2} \approx 0.32$. Therefore, $\beta = 0.5$ implies that dropping a child is always more important than moving up the tree and that for $\beta = 0.75$ it's more than twice as important to drop a child than to move from depth 3 to depth 2. Since it hard to support children in this space, it is not surprising that $k = 8$ and $k = 32$ perform relatively poorly. However, even here, we see that the gap between $k = 2$ and the other settings diminishes quickly and that for a range of $\delta$, $k = 32$ outperforms $k = 2$.

### 5.8.2 Alternative Response Functions

The analysis above uses a linear reaction curve. This embodies the assumption that all load is equally harmful and noticed. In some circumstances, a more appropriate shape could be one where a small amount of load goes unnoticed but even a moderate amount of load causes alarm and reaction. This is better modeled by a sigmoid function.

In this section, we use a parameterized sigmoid function:

$$r(x) = 1 - \frac{1}{1 + exp(\omega(x - .5))}$$

[9] Note that the scales have been adjusted as appropriate to provide a clearer view of the data.

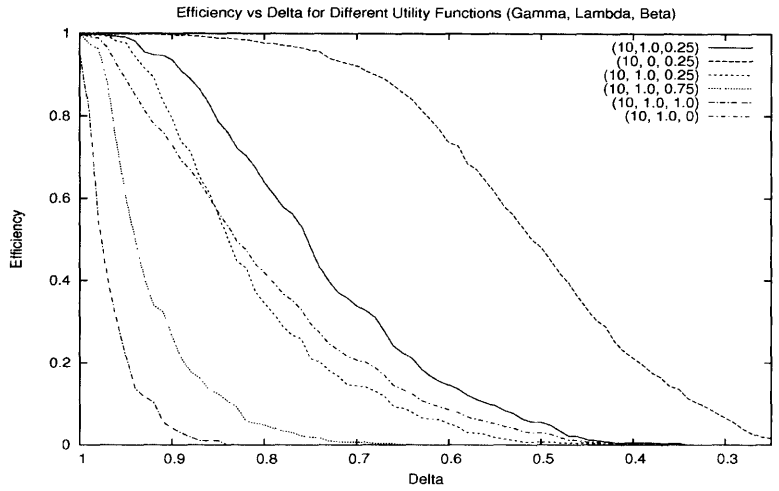



Figure 5-13: The relationship between efficiency and delta for N=100. Each line is represented by a ($\gamma$, $\lambda$, $\beta$) tuple.

Figure 5-15 depicts the sigmoid function for $\omega \in \{10, 25, 50\}$ and the linear function. As can be seen the $\omega$ parameter serves to modulate the slope of the function.

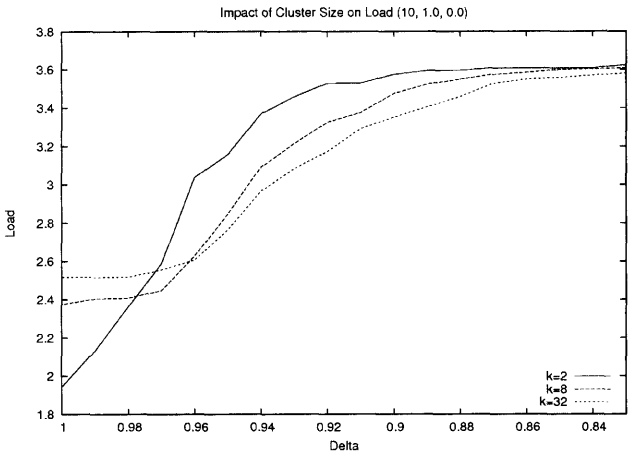
Figure 5-16 plots efficiency versus delta. Here we see two effects of the sigmoid. The flat, initial, region of the response function causes an increased amount of initial cheating. However, the sharp drop in the probability helps to induce cooperation and faithfulness. As $\omega$ increases, both of these factors become more prominent.

### 5.8.3 Information and Noise

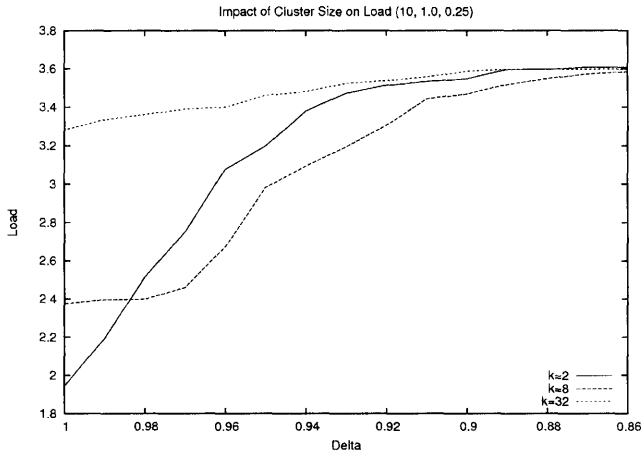
There are two key assumptions about information in the simulator. First, we assume that the nodes have perfect information about the current level of efficiency in the network. Second, we assume that the nodes have perfect understanding of the reaction curve. In practice, it is far more reasonable to assume that instead the nodes will have a noisy signal of both.

To implement such noise in the simulator, we introduce a noise term, $\epsilon \in [0, 1]$. Instead of assuming that the load signal, $\ell$ is correct, the agent treats the signal as noisy and instead assumes that the true load, $\hat{l} \in U(\ell - \epsilon\Delta, \ell + \epsilon\Delta)$ where $U(.)$ is the uniform distribution and $\Delta = L(U) - L(F)$, the improvement over unicast provided by faithful users. Therefore, when evaluating the current and/or potential scenarios, the agent will take the expected value over this range.[10] While other noise models

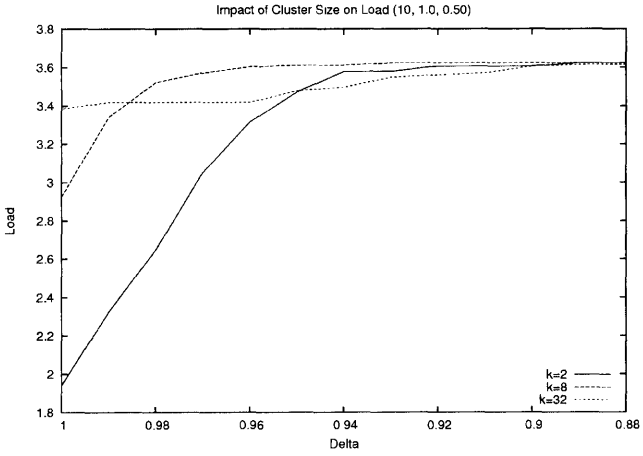
[10]There are two important implementation notes. First, the expected value is calculating by evaluating the given function at a discrete number of points in the range and averaging. Second,



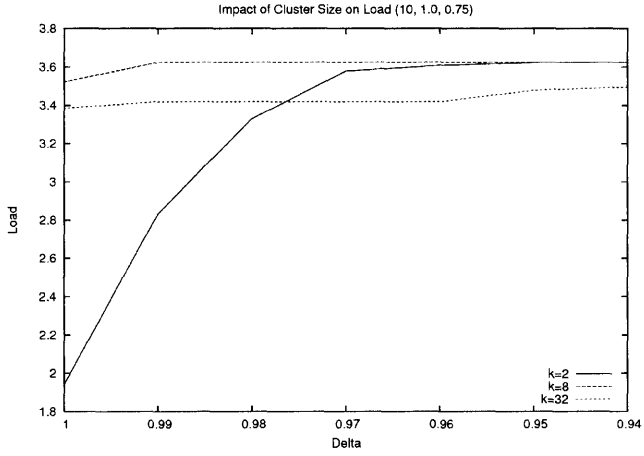

(a) $(\gamma, \lambda, \beta) = (10.0, 1.0, 0.0)$




(b) $(\gamma, \lambda, \beta) = (10.0, 1.0, 0.25)$




(c) $(\gamma, \lambda, \beta) = (10.0, 1.0, 0.5)$




(d) $(\gamma, \lambda, \beta) = (10.0, 1.0, 0.75)$

Figure 5-14: Sensitivity of k with Alternative Utility Functions

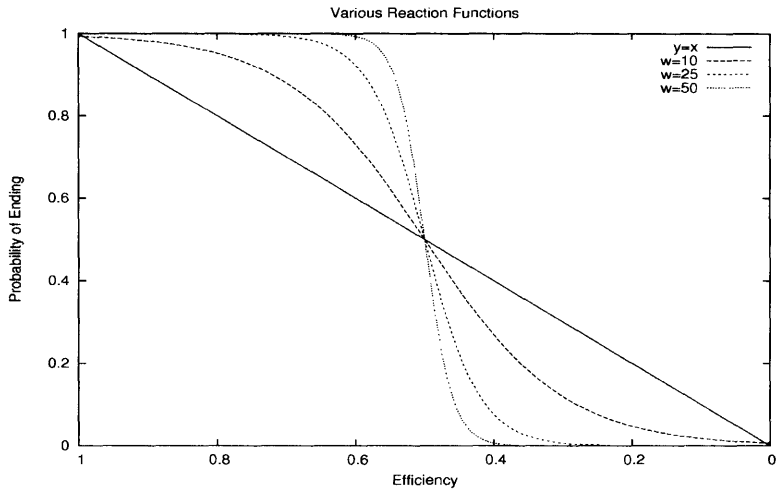



Figure 5-15: Multiple Plausible Response Functions – A linear curve and 3 Sigmoid Functions with Different $\omega$

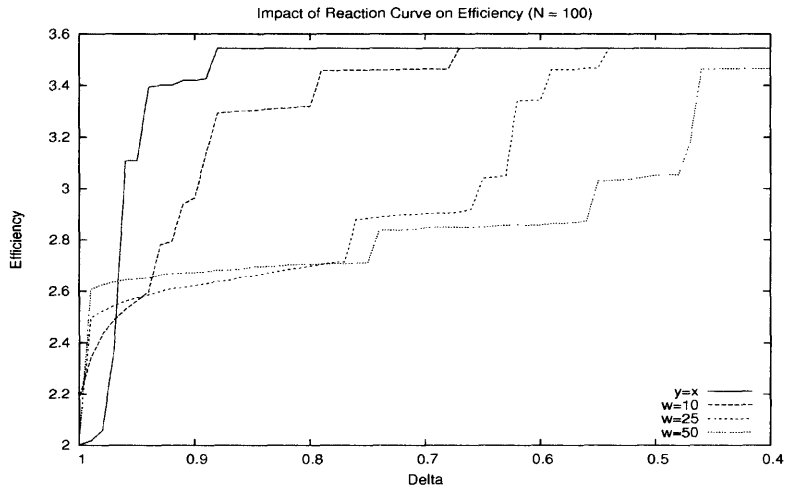
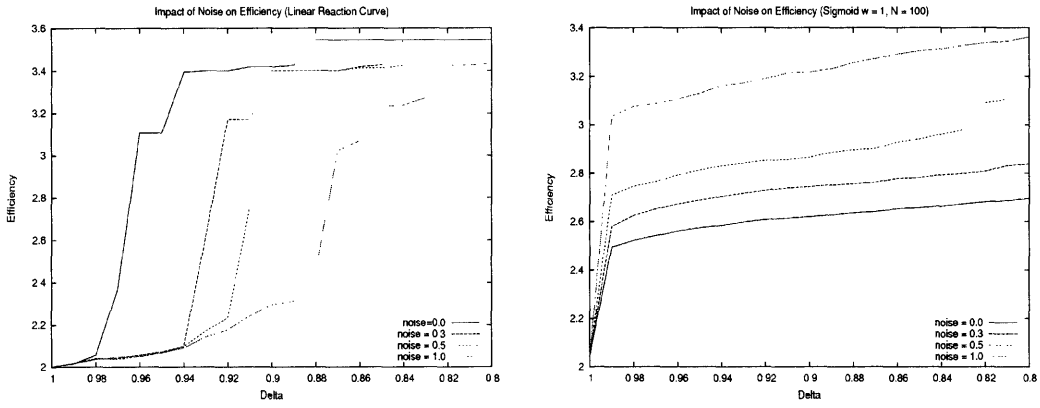



Figure 5-16: The Impact of Different Response Functions on Efficiency and Robustness

could be used, this model captures uncertainty both in the current position and in the exact nature of the reaction curve.

In this model, the impact of noise varies on the reaction function, but does not qualitatively change the results. In Figure 5-17(a), we see that with the linear response function, increased noise causes users to be more cautious – and thus more faithful. By contrast, in Figure 5-17(b) we see that the increased noise diminishes the

only points in the range [L(Faithful), L(UNI)] are considered. For points outside that range, the nearest end-point of the range is used.




(a) Linear Response Function

(b) Sigmoid Response Function ($\omega = 25$)

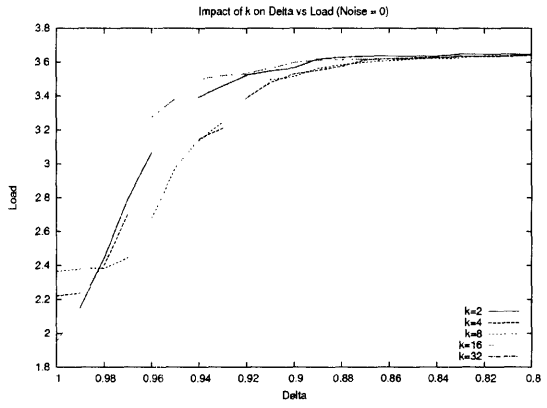
Figure 5-17: Impact of Noise on Different Response Functions

impact of the sigmoid. This should be expected as the expect value has the effect of smoothing out the sharp drop.

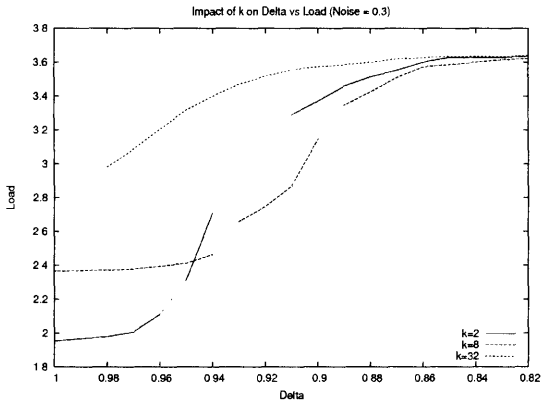
We also consider the impact of $k$ as a function of noise and see that the fundamental relationships still hold, as depicted in Figure 5-18.

## 5.9 Conclusions

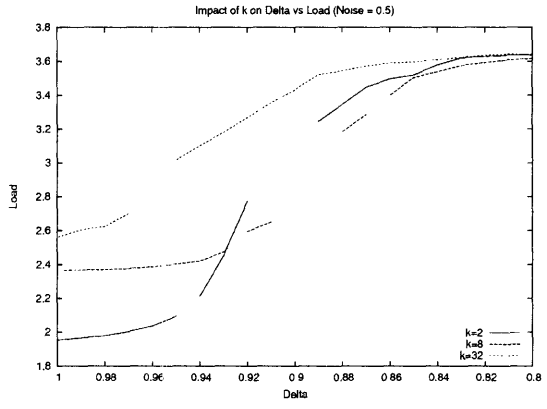
Application overlay multicast has the potential to be a lightweight alternative to IP multicast. In order to design robust overlays we need to take into account the users' incentives. In this chapter, we have described an approach to analyzing user incentives using repeated games. The critical feature of our model is that it captures the endogenous, system-induced, motivation for users to cooperate. This allowed us to identify system features and protocol parameters that can increase the robustness to selfish users. We analyzed the NICE protocol, and found that its layered branching structure helps to make the protocol more robust to selfish users; further, under reasonable assumptions, the cluster-size parameter can be used to make the protocol even more robust.



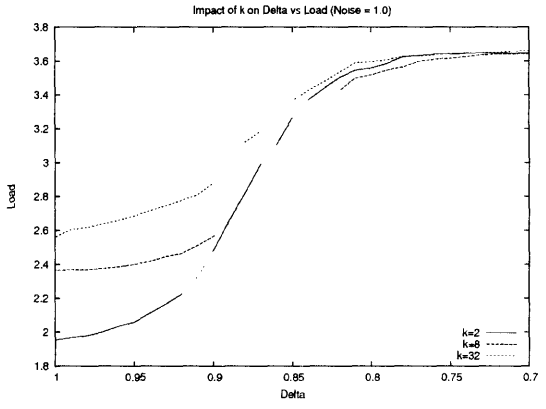

(a) $\epsilon = 0$




(b) $\epsilon = 0.3$




(c) $\epsilon = 0.5$




(d) $\epsilon = 1.0$

Figure 5-18: Impact of $k$ with Different Levels of Noise

# Chapter 6

# Discussion and Future Work

Taken together, the analysis and results of Chapters 3, 4, and 5 develop both the understanding of the particular problem domains and the understanding of repeated games as a general tool for networking problems. While each example examines a unique problem and explores the analytical approach, there are several common themes. Viewed together, these hopefully motivate further examination of repeated games as a practical tool, as well as further work on the particular problems within the frameworks established.

## 6.1 Key Contributions

**The key contribution of this thesis is the presentation of repeated games as an important and practical tool for the design of networking protocols.** This thesis is supported in two ways. The individual examples demonstrate that repeated games can provide important insight and practical results. Further, the Introduction argues that the importance of the repeated game stems from *fundamental* properties of networking applications. This suggests a broader scope and importance – and I believe that as a tool it can, should, and hopefully will be used in the future on a variety of additional problems.

This thesis also makes significant contributions within each of the problem domains considered. In Chapter 3, we demonstrate that user-directed routing has transformed routing to the point that the repeated game model is the most appropriate. This motivates us to consider coupling routing and pricing mechanisms and we discuss principles (such as using routes as the good) for doing so. In Chapter 4, we see that the desirable properties of the FPSS model do not hold in the repeated setting.

We model the repeated game and find that specific protocol parameters including the protocol period, the granularity of the format, and the width of the price field can have a significant on the outcome of the system. We show that these results are robust to a number of practical assumptions and show how these parameters can constrain the impact of the repeated dynamic, to the extent desires. Finally, in Chapter 5, we formally show that issues of user incentives present a fundamental challenge to network efficiency, formally proving this result for networks with NAT boxes. Our repeated model balances a user's immediate gratification with her (selfish) desire to see the network exist in the future. In contrast to prior models, the user's incentive to cooperate is therefore endogenous to the model, and a cooperative equilibrium does not require heavyweight mechanisms such as payment or identity systems. Using this model and our simulator, we identify the tree shape in general and the cluster size in particular for NICE as significant in building robust trees.

## 6.2 Thoughts on Repeated Games

I believe that repeated games can, and likely must, play a prominent role if game theory is to significantly impact networking research. There are several reasons for this. First and foremost, networked interactions are repeated and elementary game theory teaches us that the outcome of the one-shot game and the outcome of the repeated game can differ qualitatively. As seen in the examples, considering the repeated game exposes important and often vital dynamics and relationships.

Introducing the repeated game is not without downside. Immediately, it introduces complexity to the model. Further, it demonstrates the limitations of game theory's predictive powers inasmuch as (per the Folk Theorem) many more equilibria are possible in the repeated setting.

This thesis, however, demonstrates that these downsides can be outweighed by significant benefits. To the extent that the repeated equilibria is dramatically different from the one-shot game (or potentially so), the added complexity and uncertainty are unfortunate but perhaps necessary. It is clearly better to be uncertain about a system than to feel certain about a false conclusion. More important, this thesis demonstrates that analysis of the repeated game model – at times with appropriate assumptions – can produce conclusions which not only constrain this uncertainty but also provide practical prescriptive guidance on how to control and/or improve the system. For example, in Chapter 4, we show how seemingly benign protocol parameters can constrain the equilibrium space. Another example is Chapter 5,

where we are able to compare overlay architectures and discern the impact of the cluster size.

As discussed in the Introduction, I feel that the success found in these examples is not unique to these problems but rather related to the fundamental synergies between repeated games and networking applications. One clear instance of this is the discrete and parameterized nature of the action space in networked applications (Principle 4 from the Introduction) and its clear synergy with the significance of the action space in repeated games. Another relationship stems from the constraints inherent in networked applications (Principle 3). Naturally, invoking repeated games does not cause one to recognize or adopt these constraints to a greater degree. However, repeated games represent an unavoidable reality (e.g., in the case of routing) or a potentially simpler explanation (e.g., in the case of the application overlay) when the system's constraints preclude the realization of a simple one-shot mechanism.

Another potential source of practical results is that repeated game models can take properties, captured as fixed *exogenous* types in one-shot games, and make them *endogenous*. Clearly, in a complete model of any of the problems considered, it is important to recognize the heterogeneity of users and the existence of external types. However, our repeated models focus on endogenous motivations for important factors. In the case of routing, the problem in Chapter 3 stems from the dependency of network business relationships on the traffic flows. In contrast to the Gao and Rexford models (among others), our approach is to make these business relationships endogenous to the model. Doing so exposes the important tension between user-directed routing and commercial networks. As discussed in Chapter 5, prior models of multicast overlay networks model users as having an exogenous type (cheater or not). Instead, in our model, the user's propensity to cheat is defined by the system.

While both approaches capture correct properties, the repeated model has two strengths. First, it is a simpler model with fewer degrees of freedom. It does not require some external distribution or setting of types. As such, the model may be cleaner. A second strength is that by making the properties endogenous, it more readily facilitates practical analysis. For example, in our model of multicast overlays we are able to ask the question "How will this protocol change impact a user's desire to cheat?" This is not a question that could be asked if the propensity to cheat were a fixed exogenous type. In general, by making these motivations endogenous, the model allows for analysis of the impact of changing the relevant portion of the system (e.g., protocol, network, topology). This suggests that these models are more likely to produce practical results.

While I believe these results obtained from these models are very informative and desirable, they do contrast with style of results found in much of the literature – a difference some may find undesirable. In particular, much of the literature has focused on developing strategyproof mechanisms. Unlike the FPSS result, for example, the work on routing in this thesis does not present a way of restoring the strategyproof properties of the system in the repeated game. Instead we suggest that such a goal may be very difficult to achieve and perhaps even undesirable inasmuch as there is an underlying tussle regarding mechanism selection. Similarly, for the overlay example, our conclusions are perhaps more pessimistic than results based on a clean payment or identity system. This weakness comes from these restrictive assumptions, not the use of repeated games *per se*. That repeated games still provides practical conclusions in spite of these pessimistic assumptions should be considered a strength of the tool. Nonetheless, in both the routing and multicast application overlay problems, we are able to develop a novel understanding of the problem and find novel ways of controlling the system. To the extent that our assumptions about what is permissible and practical are correct, I feel that these conclusions while weaker in theory are stronger in practice.

One area not explored in detail in this thesis is the limitations of repeated games. The mathematics and structure of repeated games are, in general, applicable to any problem that has some element of concern about incentives and repetition. It is not however always the case that this analysis will yield interesting conclusions. Elementary game theory and our experiences presented in this thesis provide some guidance in understanding when repeated games will and will not be useful, but this is still an open (and very interesting) research question. I discuss this in more detail in the Section 6.4 below.

## 6.3 A User's Manual for Repeated Games

One question suggested by this research is "How should one use repeated games effectively as a tool?" There is no simple answer to this question. However, this thesis provides some high-level guidance and lessons.

1. *Recognize the importance of repetition in the application.* To some degree, this is the most important step. In some cases, such as our analysis of routing, this means considering a whole new aspect of the problem. In other cases, (e.g., Dellarocas' examination of reputation mechanisms, the progression from the

arguments of Crowcroft *et al* to our work on overlays), this means formalizing intuition which may be implicit in the description of the problem. Nonetheless, this thesis clearly demonstrates that repetition must be recognized to truly understand a problem. For this repetition to be meaningful, the application must be used over time and users must care to some degree about the future. Conversely, if the application is one that is truly one-shot and/or there are not incentive concerns, then repeated games will not be a useful tool. While such cases are rare, such a negative conclusion can also be useful.

2. *Develop a practical model.* To apply repeated game theory to a networked application, one must develop an appropriate model of the game. Some of the steps in the process are standard to any game theoretic setting – for example, capturing the user's utility function and action space. Recall that the user's utility function need not be complex – since it is defined based on the payoff of the stage game, not the whole game. The model will naturally and implicitly factor in the importance of the repeated context. For example, the stage-game utility function in the overlay example does not factor in any concern for the overall network. The utility function is however one case where some concerns could be raised. If the players are firms, it may be reasonable to use a simple profit function. When the players are users, it is impossible to claim that the function appropriately models the users. Instead, the goal should be to analyze a class of functions for which the function shape is likely be representative of real end-users.

   One interesting modeling step for networked applications is the information model. This must capture, for example, what users know about each other and the network in general. In networked applications it is not always clear what users (or even a central authority) can observe; further, this can differ between games. For example, in the case of routing we assume strong identities, whereas in the overlay example we do not.

   Similarly a practical model of the application should be clear on the relevant constraints – namely what tools and mechanisms are *not* permissible. For example, we do not try to force or otherwise trick ISPs into being truthful with a clever traffic-splitting routing scheme since doing so would likely violate several properties desirable for a routing system. Similarly, for the overlay example we explicitly assume the presence of NAT boxes and do not use payment schemes. These assumptions, while eventually cast into a formal game theoretic model,

have little to do with game theory and everything to do with sound system design. A model will inherently overlook certain details. However, practitioners should not shy away from being firm about what is and is not possible in practice. This thesis shows that even in the face of significant constraints, repeated games may be a useful and practical tool.

3. *Ask the questions the model is likely to answer.* Neither game theory nor repeated game theory are a magical tool, capable of addressing all incentive problems. While it is always good to look for strong conclusions and completely solutions, understanding and prescriptive advice are also very valuable in practice.

   Given a repeated game model, the first question should be "What types of equilibria are possible?" Does the outcome meaningfully differ from the one-shot game? Are these differences clearly desirable, clearly undesirable, or perhaps tussle-forming? These answers were readily available with our repeated game models. In fact, some might argue that this class of conclusions could be derived even without a formal model. While this may be the case, developing the formal model need not be very time consuming, provides more confidence in the conclusions, and of course facilitates further analysis.

   The second question that should be asked is "How do design decisions and key parameters impact the set of equilibria?" In the case of routing, we examine this question analytically, by taking partial derivatives. In the overlay example, we use simulation to better understand the interplay of these variables on synthetic network topologies. Understanding the impact of these factors can help to address the equilibrium space explosion that is often suffered when going to a repeated game model.

## 6.4 Open Questions and Future Research

I hope that this thesis represents a step forward in the use of repeated games and the examination of the particular problems. There are several aspects of the general thesis and particular problems that I believe are grounds for interesting and relevant future work.

### 6.4.1 Repeated Games

I believe that there are two very interesting open high-level questions:

- *"What is an appropriate equilibrium notion for repeated games?"* One very important and interesting area for future research is that of equilibrium notions and refinement for repeated games. Chapter 4, discusses the impact of the equilibrium notion in several places, most particularly Section 4.6.2. Here the results generalize, but the question is how to come up with a defensible equilibrium refinement that motivates a reasonable class of strategies, such as the class of Proportional Punishment strategies. The problem here is balancing the tension between over-constraining the equilibria (and thus not permitting the cooperation) and under-constraining the equilibria (and thus permitting the grim strategy – or something analogous). In Chapter 5, the discussion of the equilibrium notion was more implicit in the formulation of the problem – and the complexity of strategies and reasoning that was assumed. It would be perhaps more realistic to assume that players consider the reactions of other players – and doing so would only strengthen the results. It is however unclear how to reason about this process however and formulate the dynamics.

- *"When does one **not** need to worry about the repeated game?"* A more formal understanding of this question could be very useful and interesting. This would be useful not only to limit needless work, but also to provide formal guarantees that a mechanism or approach will be robust under repetition. Here the fundamental obstacle is the Folk Theorem, which allows all individually rational outcomes in the stage-game to be enforceable equilibria of the repeated game. Perhaps however, there are reasonable assumptions to make in the case of a networked application, recognizing the various constraints on mechanisms and strategies that could limit the set of equilibria in a meaningful way that is generalizable. An informal analysis of the problem suggests starting points – large numbers of players, certain non-cooperative game structures, etc. Formalizing this could be both interesting and useful.

### 6.4.2 Individual Problems

I believe that there is much progress to be made by looking at the practical reality of routing. One important observation is that the preferences of networks are driven by the fact that they are profit maximizing firms. Already, we have seen this approach

informing our research on routing [42, 97]. However, much routing research still focusing on the ranking of routes and the preferences expressed through such rankings. In many cases, the preferences – while manifest in a complex ranking – derive from a very simple logic. Another important observation is that routing today is, through the amount of ranking, filtering, and modifying of information, essentially a bilateral protocol between networks: the information exchanged between two networks need not directly influence the information that is then passed on.

In the context of the particular problem of the Repeated Incentive Routing Game, presented in Chapter 4, there are a variety of relaxations that could be made to the model. Many of these are discussed in the chapter, particularly in Section 4.6.2. Perhaps the most interesting would be to examine more general networks. In equilibrium, one might expect the results to be the same. However, appropriate models of uncertainty in future costs, especially in the case of asynchronous play, could be interesting and require interesting models of the hierarchy of AS relationships.

In the case of application multicast overlay, several interesting extensions are possible. One would be to consider some of the alternative motivations presented in the chapter for cooperation. In particular, the notion of users leaving after degrading quality seems particularly interesting as it may cause the strategy space to become more involved. Another interesting area would be the selective application of some of the mechanisms that were presented as incomplete solutions. For example, payments cannot solve the general problem. However, applied in particular instances (e.g., large corporations paying for better quality), they may be useful. Understanding when and how to apply these – and how they help the repeated dynamic to support larger equilibria could be a fascinating and pragmatic extension to the work. Lastly, examination of additional overlay protocols and examination of the behavior of currently deployed systems also may yield interesting results.

### 6.4.3 Addition Problems

One area of future work which I believe holds great promise is the application of repeated game analysis to other problems not directly considered in this thesis. These problems can involve firms (e.g., additional network competition, competition for wireless coverage, web services, or caching), individuals (e.g., ad-hoc networks, other peer-to-peer problems), or both (e.g., grid networks, sensor networks). In all of these problems, the inter-temporal dynamic has the potential to play a significant role, suggesting that considering the problem in a repeated context may produce important

novel insights.

Each of these problems has the potential to draw upon – but moreover build upon – the repeated game tools and frameworks presented in this thesis. For example, while each problem differs, it is likely that the problems involving firms will map somewhat to the routing examples, with firms trading off near-term profits for long-term cooperation. Some of the more individual examples may map more closely to the multicast example. For example, in the case of ad-hoc networks, cooperation comes at a cost but helps to ensure the viability of the network. However, what will likely be more interesting are the aspects of the analysis which diverge from the the analysis presented in this thesis. These results can be meaningful in their own right, but moreover increase the strength and applicability of repeated game analysis as a tool for networked application and protocol design.

# Bibliography


[1] Mike Afergan and John Wroclawski. On the Benefits and Feasibility of Incentive Based Routing Infrastructure. In *PINS '04: Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, pages 197–204. ACM Press, 2004.

[2] Akamai: Facts and Figures. http://www.akamai.com/en/html/about/facts_figures.html.

[3] Akamai Technologies Homepage. http://www.akmai.com/.

[4] K. Almeroth. The Evolution of Multicast: From the MBone to Inter-Domain Multicast to Internet2 Deployment. *IEEE Network*, January 2000.

[5] Dave Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP)*, 2001.

[6] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy. Scalable application layer multicast. In *Proceedings of the 2002 Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 205–217. ACM Press, 2002.

[7] Joseph Bertrand. Théorie Mathématique de la Richesse Sociale. *Journal des Savants*, pages 499–508, 1883.

[8] Robert Beverly. Reorganization in Network Regions for Optimality and Fairness. Master's thesis, MIT, August 2004.

[9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), December 1998. Updated by RFC 3260.

[10] Cache Logic - P2P Traffic Analysis. http://www.cachelogic.com/research/slide3.php.

[11] 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.

[12] N. Christin and J. Chuang. A Cost-Based Analysis of Overlay Routing Geometries. In *Proceedings of IEEE INFOCOM 2005*, Miami, FL, March 2005.

[13] Nicolas Christin, Jens Grossklags, and John Chuang. Near Rationality and Competitive Equilibria in Networked Systems. In *PINS '04: Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, pages 213–219, New York, NY, USA, 2004. ACM Press.

[14] Yang-Hua Chu, Aditya Ganjam, T. S. Eugene Ng, Sanjay G. Rao, Kunwadee Sripanidkulchai, Jibin Zhan, and Hui Zhang. Early Experience with an Internet Broadcast System Based on Overlay Multicast. In *USENIX Annual Technical Conference, General Track*, pages 155–170, 2004.

[15] Yang-Hua Chu, Sanjay G. Rao, and Hui Zhang. A Case for End System Multicast. In *ACM SIGMETRICS 2000*, pages 1–12, Santa Clara, CA, June 2000. ACM.

[16] B. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz. Characterizing Selfishly Constructed Overlay Networks, 2004.

[17] David D. Clark, John Wroclawski, Karen R. Sollins, and Robert Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. In *Proceedings of ACM SIGCOMM 2002*, Pittsburgh, PA, August 2002.

[18] D.D. Clark. Policy routing in Internet protocols. RFC 1102, May 1989.

[19] Jacomo Corbo and David C. Parkes. The Price of Selfish Behavior in Bilateral Network Formation. In *PODC '05: Proceedings of the 24rd ACM Symposium on Principles of Distributed Computing*, Las Vegas, Nevada, USA, 2005.

[20] Augustin Cournot. Recherches sur les Principles Mathematiques de la Theorie des Richesses. *Researches into the Mathematical Principles of the Theory of Wealth*, 1838.

[21] Landon P. Cox and Brian D. Noble. Samsara: Honor Among Thieves in Peer-to-Peer Storage. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pages 120–132, Bolton Landing, NY, USA, October 2003.

[22] Peter Cramton. Collusive Bidding: Lessons from the FCC Spectrum Auctions. *Journal of Regulatory Economics*, 17:229–252, 2000.

[23] Stephen Edward Deering. *Multicast routing in a datagram internetwork.* PhD thesis, Stanford, CA, USA, 1992.

[24] Chrysanthos Dellarocas. Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior. In *EC '00: Proceedings of the 2nd ACM Conference on Electronic commerce*, pages 150–157, New York, NY, USA, 2000. ACM Press.

[25] Chrysanthos Dellarocas. Analyzing the Economic Efficiency of eBay-like Online Reputation Reporting Mechanisms. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 171–179. ACM Press, 2001.

[26] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment Issues for the IP Multicast Service and Architecture. *IEEE Network*, pages 78–88, Jan./Feb. 2000.

[27] G. Ellison. Cooperation in the Prisioner's Dilemma with Anonymous Random Matching. In *Review of Economics Studies 61*, pages 567–588, 1993.

[28] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. RFC 2362 (Experimental), June 1998.

[29] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *PODC '03: Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing*, pages 347–351, New York, NY, USA, 2003. ACM Press.

[30] Nick Feamster. Practical Verification Techniques for Wide-Area Routing. *SIGCOMM Computer Communication Review*, 34(1):87–92, 2004.

[31] Nick Feamster, Hari Balakrishnan, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. The Case for Separating Routing from Routers. In

*Procedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, August 2004.

[32] Joan Feigenbaum, Christos Papadimitriou, Rahul Sami, and Scott Shenker. A BGP-Based Mechanism for Lowest-Cost Routing. In *PODC '02: Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing*, pages 173–182, New York, NY, July 2002.

[33] Joan Feigenbaum, Christos H. Papadimitriou, and Scott Shenker. Sharing the Cost of Multicast Transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.

[34] Joan Feigenbaum, Rahul Sami, and Scott Shenker. Mechanism Design for Policy Routing. In *PODC '04: Proceedings of the 23rd Annual Symposium on Principles of Distributed Computing*, pages 11–20, New York, NY, USA, 2004. ACM Press.

[35] 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(DIALM)*, pages 1–13, New York, NY, USA, 2002. ACM Press.

[36] Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang. Robust Incentive Techniques for Peer-to-Peer Networks. In *EC '04: Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 102–111. ACM Press, 2004.

[37] Michal Feldman, Christos Papadimitriou, John Chuang, and Ion Stoica. Free-riding and Whitewashing in Peer-to-Peer Systems. In *PINS '04: Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, pages 228–236. ACM Press, 2004.

[38] D. Fudenberg and D. Levine. Efficiency and Observability in Games with Long-Run and Short-Run Players. *Journal of Economic Theory 62*, pages 103–135, 1994.

[39] D. Fudenberg, D. Levine, and E. Maskin. The Folk Theorem in Repeated Games with Imperfect Public Information. *Econometrica 62*, pages 997–1039, 1994.

[40] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, 1991.

[41] Lixin Gao, Timothy Griffin, and Jennifer Rexford. Inherently safe backup routing with BGP. In *Proceedings of INFOCOM 2001*, pages 547–556, 2001.

[42] Lixin Gao and Jennifer Rexford. Stable Internet Routing Without Global Coordination. In *Measurement and Modeling of Computer Systems*, 2000.

[43] David K. Goldenberg, Lili Qiuy, Haiyong Xie, Yang Richard Yang, and Yin Zhang. Optimizing Cost and Performance for Multihoming. In *Proceedings of ACM SIGCOMM 2004*, pages 79–92. ACM Press, 2004.

[44] Edward J. Green and Robert H. Porter. Noncooperative Collusion under Imperfect Price Information. *Econometrica*, (1):87–100, Jan 1984.

[45] J. Hershberger and S. Suri. Vickrey Prices and Shortest Paths: What is an Edge Worth? In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, page 252. IEEE Computer Society, 2001.

[46] Yang hua Chu, John Chuang, and Hui Zhang. A Case for Taxation in Peer-to-Peer Streaming Broadcast. In *PINS '04: Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, pages 205–212, New York, NY, USA, 2004. ACM Press.

[47] Yang hua Chu and Hui Zhang. Considering Altruism in Peer-to-Peer Internet Streaming Broadcast. In *NOSSDAV '04: Proceedings of the 14th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 10–15, New York, NY, USA, 2004. ACM Press.

[48] Elgan Huang, Jon Crowcroft, and Ian Wassell. Rethinking Incentives for Mobile Ad Hoc Networks. In *PINS '04: Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, pages 191–196, New York, NY, USA, 2004. ACM Press.

[49] Interpret the Destination Class Usage MIB. http://www.juniper.net/techpubs/software/junos/junos56/swconfig56-net-mgmt/html/dcu-mib.html.

[50] IP Multicast - Cisco Systems. http://www.cisco.com/en/US/tech/tk828/tsd_technology_support_protocol_home.html.

[51] M. O. Jackson. A Survey of Models of Network Formation: Stability and Efficiency. Game Theory and Information 0303011, Economics Working Paper Archive at WUSTL, March 2003. Available at http://ideas.repec.org/p/wpa/wuwpga/0303011.html.

[52] John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole, Jr. Overcast: Reliable Multicasting with an Overlay Network. pages 197–212.

[53] Ramesh Johari and John N. Tsitsiklis. Efficiency Loss in a Network Resource Allocation Game. *Mathetmatics of Operations Research*, 29(3):407–435, 2004.

[54] III John W. Stewart. *BGP4: Inter-Domain Routing in the Internet.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.

[55] Juniper Networks :: Multicast. http://www.juniper.net/products/eseries/ip_services_multicast.html.

[56] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM Press.

[57] F. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. In *Journal of the Operational Research Society*, volume 49, 1998.

[58] P. Klemperer. What Really Matters in Auction Design. *Journal of Economic Perspectives*, (1):169–189, 2002.

[59] P. Klemperer. Using and Abusing Economic Theory. *Journal of the European Economic Association*, (1), 2003.

[60] L. Mathy, R. Canonico, and D. Hutchinson. An Overlay Tree Building Control Protocol. In *International Workshop on Networked Group Communication (NGC)*, pages 76–87, Nov 2001.

[61] Roger Lagunoff and Akihiko Matsui. Asynchronous Choice in Repeated Coordination Games. *Econometrica*, 65(6):1467–1478, 1997.

[62] Seungjoon Lee, Rob Sherwood, and Samrat Bhattacharjee. Cooperative Peer Groups in NICE. In *Proceedings of INFOCOM 2003*, 2003.

[63] Lun Li, David Alderson, Walter Willinger, and John Doyle. A First-Principles Approach to Understanding the Internet's Router-level Topology. In *Proceedings of the 2004 ACM SIGCOMM*, pages 3–14, New York, NY, USA, 2004. ACM Press.

[64] LimeWire User Guide. http://www.limewire.com/english/content/ug_options.shtml.

[65] Sergio Marti and Hector Garcia-Molina. Limited Reputation Sharing in P2P Systems. In *EC '04: Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 91–101, New York, NY, USA, 2004. ACM Press.

[66] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory.* Oxford University Press, New York, 1995.

[67] Laurent Mathy, Nick Blundell, Vincent Roca, and Ayman El-Sayed. Impact of Simple Cheating in Application-Level Multicast. In *Proceedings of IEEE Infocom*, 2004.

[68] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. BRITE: Universal Topology Generation from a User's Perspective. Technical Report 2001-003, 1 2001.

[69] Modeling Topology of Internetworks Home Page. http://www.cc.gatech.edu/projects/gtitm/.

[70] Richard Mortier and Ian Pratt. Incentive Based Inter-domain Routeing. In *Proceedings of Internet Charging and QoS Technology Workshop (ICQT'03)*, pages 308–317, September 2003.

[71] J. Nash. Equilibrium Points in N-Person Games. In *Proceedings of the National Academy of Sciences*, volume 36, pages 48–49.

[72] Net Blackout Marks Web's Achilles Heel. http://news.com.com/2100-1033-267943.html?legacy=cnet.

[73] Tsuen-Wan Ngan, Dan S. Wallach, and Peter Druschel. Incentives-Compatible Peer-to-Peer Multicast. In *2nd Workshop on the Economics of Peer-to-Peer Systems*, Cambridge, Massachusetts, June 2004.

[74] Noam Nisan and Amir Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35(1):166–196, 4 2001.

[75] Andrew M. Odlyzko. Pricing and Architecture of the Internet: Historical Perspectives from Telecommunications and Transportation. http://www.dtc.umn.edu/ odlyzko/doc/pricing.architecture.pdf.

[76] Tobias Oetiker. MRTG: The Multi Router Traffic Grapher. In *Proceedings of the 12th Conference on Systems Administration*, pages 141–148. USENIX Association, 1998.

[77] M.J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, Cambridge MA, 1994.

[78] Dimitris Pendarakis, Sherlia Shi, Dinesh Verma, and Marcel Waldvogel. ALMI: An Application Level Multicast Infrastructure. In *3rd USNIX Symposium on Internet Technologies and Systems (USITS)*, pages 49–60, San Francisco, CA, USA, March 2001.

[79] Larry L. Peterson and Bruce S. Davie. *Computer Networks: A Systems Approach.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[80] Price Match Guarantee- Sears Canada Inc. http://www.sears.ca/e/customerservice/price_match.htm.

[81] Barath Raghavan and Alex C. Snoeren. A System for Authenticated Policy-Compliant Routing. In *Proceedings of ACM SIGCOMM*, September 2004.

[82] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, 1984.

[83] Vincent Roca and Ayman El-Sayed. A Host-Based Multicast (HBM) Solution for Group Communications. In *ICN '01: Proceedings of the First International Conference on Networking-Part 1*, pages 610–619, London, UK, 2001. Springer-Verlag.

[84] Michael H Rothkopf, Thomas J Teisberg, and Edward P Kahn. Why Are Vickrey Auctions Rare? *Journal of Political Economy*, 98(1):94–109, February 1990.

[85] T. Roughgarden. Designing Networks for Selfish Users is Hard. In *FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 472, Washington, DC, USA, 2001. IEEE Computer Society.

[86] Tim Roughgarden and Éva Tardos. How Bad is Selfish Routing? *J. ACM*, 49(2):236–259, 2002.

[87] Naouel Ben Salem, Jean-Pierre Hubaux, and Markus Jakobsson. Reputation-based Wi-Fi Deployment Protocols and Security Analysis. In *WMASH '04: Proceedings of the 2nd ACM international Workshop on Wireless mobile applications and services on WLAN hotspots*, pages 29–40, New York, NY, USA, 2004. ACM Press.

[88] K. Sarac and K. Almeroth. Supporting Multicast Deployment Efforts: A Survey of Tools for Multicast Monitoring. *Journal of High Speed Networking*, March 2001.

[89] Stefan Savage, Thomas Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoff Voelker, and John Zahorjan. Detour: Informed Internet Routing and Transport. *IEEE Micro*, 19(1):50–59, 1999.

[90] Stefan Savage, Andy Collins, Eric Hoffman, John Snell, and Thomas E. Anderson. The End-to-End Effects of Internet Path Selection. In *SIGCOMM*, pages 289–299, 1999.

[91] Baochun Li Selwyn Yuen. Market-driven Bandwidth Allocation in Selfish Overlay Networks. In *Proceedings of INFOCOM 2005*, 2005.

[92] Srinivas Shakkotai and R. Srikant. Economics of Network Pricing with Multiple ISPs. In *Proceedings of INFOCOM 2005*, 2005.

[93] Jeffrey Shneidman and David C. Parkes. Rationality and Self-Interest in Peer to Peer Networks. In *2nd Int. Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003.

[94] Sourceforge.net: Welcome. http://sourceforge.net/.

[95] Staples: What is Staples' price-match policy? http://www.staples.com/content/help/using/general_match.asp.

[96] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz. Characterizing the internet hierarchy from multiple vantage points, 2001.

[97] Lakshminarayanan Subramanian, Matthew Caesar Cheng Tien Ee, Mark Handley, Morley Mao, Scott Shenker, and Ion Stoica. HLP: A Next Generation Inter-domain Routing Protocol. In *Proceedings of the 2005 ACM SIGCOMM Conference*, New York, NY, USA, 2005. ACM Press.

[98] Andrew Tanenbaum. *Computer Networks.* Prentice Hall Professional Technical Reference, 2002.

[99] Renata Teixeira, Aman Shaikh, Tim Griffin, and Jennifer Rexford. Dynamics of Hot-Potato Routing in IP Networks. In *Proceedings of ACM SIGMETRICS*, 2004.

[100] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

[101] D. Waitzman, C. Partridge, and S.E. Deering. Distance Vector Multicast Routing Protocol. RFC 1075 (Experimental), November 1988.

[102] Quan Wen. Repeated games with asynchronous moves. Technical Report 0204, Department of Economics, Vanderbilt University, April 2002. Available at http://ideas.repec.org/p/van/wpaper/0204.html.

[103] Xiaowei Yang. Nira: a new Internet routing architecture. *SIGCOMM Comput. Commun. Rev.*, 33(4):301–312, 2003.

[104] Beichuan Zhang, Sugih Jamin, and Lixia Zhang. Host Multicast: A Framework for Delivering Multicast To End Users. In *Proceedings of INFOCOM 2002*, 2002.