Effective Wide-Area Network Performance Monitoring and Diagnosis from End Systems

by

Ying Zhang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2009

Doctoral Committee :
Assistant Professor Zhuoqing Mao, Chair
Professor Farnam Jahanian
Assistant Professor Jason Nelson Flinn
Assistant Professor Clayton D Scott

To my family.

ACKNOWLEDGEMENTS

This dissertation could not have been completed without the support and encouragement of many people. First, I would like to thank my advisor, Professor Zhuoqing Morley Mao. Morley is a great advisor. I have learned a great deal about research, academic writing and presentation skills from her. She always encourages me to think independently and discuss with her any research ideas. Working with Morley, I have not only developed professional research skills in computer networks but also have deepened my research interests in general system area, such as security and operating system. I am very lucky to have worked with her as a student, teaching assistant and research assistant for the past 5 years.

I would also like to thank my dissertation committee, Professor Jason Flinn, Professor Farnam Jahanian and Professor Clayton Scott, for their time and effort to help improve and refine my thesis.

I appreciate Dr. Ming Zhang's mentorship and support throughout my internship at Microsoft Research. He has provided valuable suggestions to both my thesis work and the summer project in Microsoft. I also appreciate many researchers in AT&T Labs for the opportunities of internship and the discussions at the early stage of my graduate studies.

The five-year life in graduate school has become wonderful because of many colleagues and friends. I would like to thank all the students in our group: Xu (Simon) Chen, Zhiyun Qian, Feng Qian and Yudong Gao, Ni Pan, and Birjodh Tiwana. I would also like to thank other friends in the department for valuable discussions and suggestions on my work: Kevin Borders, Sushant Sinha, Kaushik Veeraraghavan, Ya-Yunn Su, and Evan Cooke.

Finally, my thanks goes to my parents, Xiaogang Zhang and Xiaolin Lu, who have always been a constant source of love and inspiration. I am very lucky to have so many wonderful people always behind me. I dedicated this dissertation to them.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES ix

LIST OF TABLES xii

CHAPTERS

I Introduction 1

II Background and Related Work 8
- 2.1 Introduction to BGP 8
  - 2.1.1 BGP basic operations 9
  - 2.1.2 BGP dynamics 11
  - 2.1.3 BGP security issues 13
- 2.2 Characterization of Internet control plane 15
  - 2.2.1 Network topology discovery 16
  - 2.2.2 BGP diagnosis and root cause analysis 18
- 2.3 Characterization of Internet data plane 19
  - 2.3.1 Data-plane measurement methodology 20
  - 2.3.2 Topology discovery 22
  - 2.3.3 Network tomography 23
- 2.4 Interaction between control and data plane 24
- 2.5 Infrastructure-based monitoring and diagnosis 26
  - 2.5.1 SLA monitoring 26
  - 2.5.2 Network diagnosis from ISPs' perspective 28
- 2.6 End-host based monitoring and diagnosis 29

III On the Impact of Route Monitor Selection 31
- 3.1 Introduction 31
- 3.2 Methodology 33
  - 3.2.1 Route monitor locations 33

3.2.2 Discovery of static network properties 35
3.2.3 Discovery of dynamic network properties 36
3.2.4 Inference of network properties 37
3.3 Deployment scenario analysis 38
3.4 Monitor Selection Analysis 41
3.4.1 Monitor selection schemes 41
3.4.2 Discovery of static network properties 42
3.4.3 Discovery of dynamic network properties 44
3.4.4 Inference of network properties 46
3.5 Summary 50

IV Diagnosing Routing Disruptions from End Systems 53
4.1 Introduction 53
4.2 System Architecture 57
4.3 Collaborative Probing 59
4.3.1 Learning routing state via probing 59
4.3.2 Discussion 61
4.4 Event Identification and Classification 63
4.5 Event Correlation and Inference 67
4.5.1 Inference model 68
4.5.2 Inference algorithm 72
4.6 Results of Event Identification and Classification 74
4.6.1 Data cleaning process 76
4.6.2 Identified events and their classification 77
4.6.3 Validation with BGP data 79
4.7 Results of Event Correlation and Inference 81
4.7.1 Result summary 81
4.7.2 Validation with BGP-based inference for a Tier-1 ISP 84
4.7.3 Validation with BGP-based inference and Syslog analysis for Abilene 88
4.7.4 Validation with NANOG mailing list 89
4.8 Performance Impact Analysis 92
4.9 System Evaluation 94
4.10 Summary 95

V Detecting Traffic Differentiation in Backbone ISPs 98
5.1 Introduction 98
5.2 Traffic differentiation 102
5.3 Methodology 104
5.3.1 Path selection 104
5.3.2 Loss rate measurement 107
5.3.3 Differentiation detection 109
5.4 Implementation 112
5.5 Eliminating noise effects 114
5.5.1 Overloaded probers 115

5.5.2 ICMP rate limiting 116
5.5.3 Loss on reverse path 117
5.5.4 Effects of load balancing 119
5.6 Experimental results 120
5.6.1 Content-based differentiation 120
5.6.2 Validation with two-ended controlled probing 122
5.6.3 Routing-based differentiation 123
5.6.4 Correlation with TOS value 126
5.6.5 Correlation with network load 129
5.6.6 Degree of differentiation 131
5.6.7 Implementation of differentiation in router testbed . . 132
5.7 System evaluation 135
5.8 Summary 137

VI Measuring and Predicting the Impact of Routing Changes 140
6.1 Overview 140
6.2 Measurement Methodology 144
6.2.1 Terminology 144
6.2.2 Data Collection 145
6.2.3 Active Probing Methodology 145
6.3 Experiment setup 148
6.3.1 Data Collection 148
6.3.2 Probing Control 148
6.3.3 Probing System Performance 149
6.3.4 Probing System Limitations 150
6.4 Characterizing Data Plane Failures 151
6.4.1 Overall Statistics 151
6.4.2 Reachability Failures 152
6.4.3 Forwarding Loops 156
6.5 Failure Prediction Model 160
6.5.1 Prediction Model 161
6.6 Summary 166

VIIComparing Backbone ISPs using Multiple Performance Metrics . . . 169
7.1 Introduction 169
7.2 Methodology 172
7.2.1 Path metrics 173
7.2.2 Path scores 179
7.2.3 Discussion 180
7.3 Implementation 181
7.4 System Evaluation 183
7.4.1 Validation 185
7.5 Results 187
7.5.1 Unreachability 189
7.5.2 Loss rate 196

7.5.3 Correlation between metrics 201
7.6 System Performance 204
7.7 Conclusion 206

VIIIConclusion 208
8.1 Thesis summary and discussion 208
8.1.1 On the impact of route monitor selection: better monitor placement 209
8.1.2 Diagnosing routing disruptions: coverage and accuracy trade-offs 209
8.1.3 Traffic differentiation: detection and prevention 211
8.1.4 Applications of end-host based monitoring systems . . 211
8.2 Future work 213
8.2.1 Monitoring in Enterprise network and data center environment 214
8.2.2 Monitoring in online social network 215
8.2.3 Analyzing the economical and technological factors in the Internet 215

BIBLIOGRAPHY 217

LIST OF FIGURES

Figure

3.1 Distribution of observed links across tiers. 39
3.2 Monitors in Tier 1 40
3.3 Monitors in Tier 2 41
3.4 Monitors in Tier 3 41
3.5 Number of observed links 42
3.6 Number of observed multi-homing stub ASes 43
3.7 Observed AS path count (including subpaths) 43
3.8 Fraction of observed routing events 44
3.9 Evasion of prefix hijacking detection: number of attacker-victim pairs 45
3.10 Evasion of prefix hijacking detection: number of attackers per victim 45
3.11 Evasion of prefix hijacking detection: number of victims per attacker 45
3.12 Profit-driven path prediction accuracy (length match). 46
3.13 Number of matched top degree AS in all observed AS paths 47
3.14 Sampled path prediction accuracy: exact matching (new algorithm) . 49
4.1 Collaborative probing to discover routing events. 58
4.2 System Architecture 58
4.3 The bipartite graph of root cause inference 70
4.4 Number of changes detected (Sep. 25, 2007) 74
4.5 Detection ratio changes with probing interval and bandwidth 78
4.6 CDF of the number of events per cluster 82
4.7 CDF of the violations per cluster 82
4.8 CDF of the confidence per cluster 82
4.9 Matching rate for hot-potato changes – a common type of routing disruption. 88
4.10 Delay change distribution in each category for AS7018 (actual path delay). 92
4.11 Comparison between absolute path delay and target delay changes (AS7018, type:new distance decrease and external AS path change) . 92

4.12 Delay change distribution of routing change across different target AS (old dist. inc, target) 92
4.13 Delay change distribution of routing change across different target AS (old edge down, target) 93
4.14 Relative execution time compared with the probing interval 95
5.1 An example of differentiation implementation. 103
5.2 Select path to discover various types of traffic differentiation. 104
5.3 The NVLens architecture 111
5.4 Impact of CPU utilization on loss rate. 116
5.5 Impact of probing frequency on loss rate. 117
5.6 Impact of probing packet size on loss rate. 118
5.7 Loss rate ratio (filtering vs. no filtering). 118
5.8 Validation using two-ended controlled measurement 122
5.9 Loss rate difference for previous-hop AS based differentiation. 131
5.10 Loss rate difference for content-based differentiation. 131
5.11 Router testbed setup 133
5.12 Loss rate difference on router testbed 133
5.13 Impact of redundancy factor. 134
5.14 Impact of the maximum probing threshold. 135
5.15 Execution time and memory usage of path selector. 135
5.16 Probing overhead under single-ISP *vs.* multi-ISP path selection. . . . 135
6.1 Active probing architecture for vantage point X (both functionalities can be implemented on the same host). 146
6.2 Probing delay distribution for each BGP feed: Most delays are within 100 seconds. 149
6.3 Destination prefixes and ASes affected by reachability problems. . . . 153
6.4 Normalized hop distance btw. the source and the last received traceroute reply. 154
6.5 Duration of unreachable incidences. 155
6.6 Appearance probability and conditional probability (conditioned on the responsible AS) of unreachable incidences. 156
6.7 Destination prefixes and ASes affected by forwarding loops. 158
6.8 Duration of loop incidences. 159
6.9 Appearance probability and conditional probability (conditioned on the responsible AS) of loop incidences. 160
6.10 Average value of Λ for each prefix 165
6.11 Receiver operating characteristics for selected subset of prefixes. . . . 166
6.12 Receiver operating characteristics for all probed prefixes. 166
7.1 Fraction of PoP-dst pairs with unreachability problem. 187
7.2 Compare unreachability for for U.S. vs. international paths. 188
7.3 Compare unreachability for short paths vs. long paths. 188
7.4 Compare unreachability to Microsoft and Google 189

7.5 Compare unreachability to Comcast and RoadRunner 189
7.6 Compare unreachability to Akamai and Limelight 189
7.7 Compare unreachability in two PoPs: New York and Los Angeles to prefix 84.38.208.0/20 (a tier-3 U.S. ISP). 190
7.8 Compare unreachability scores for two contiguous time periods. . . . 191
7.9 Compare unreachability scores for 5-month separated time periods. . 192
7.10 Compare unreachability scores between day and night. 194
7.11 Correlation of unreachability to 202.57.3.0/24 (an ISP in Indonesia) from two ISPs: strongly correlated. 195
7.12 Correlation of unreachability to 72.14.235.0/24 (YouTube prefix) from two ISPs: uncorrelated. 195
7.13 Loss rate comparison across ISPs. 196
7.14 Compare average loss rate for long vs. short paths. 197
7.15 Compare average loss rate for U.S. vs. international paths. 197
7.16 Average loss rate distribution between large PoPs. 197
7.17 Compare average loss rate for 5-month separated time periods. 201
7.18 Compare average loss rate for two contiguous time periods. 201
7.19 Time of day effect on average loss rate. 201
7.20 Correlation of average loss rate between Level3 and AT&T (from New York to Chicago) 202
7.21 Correlation between loss rate and stretch between New York and Seattle.204
7.22 Correlation between unreachability and stretch to destination 202.65.134.0/24 (a company in HK). 205
7.23 Correlation between unreachability and diversity to destination 202.88.241.0/24 (an ISP in India). 205

LIST OF TABLES

Table

3.1 Comparison among three deployment scenarios. 38

3.2 Statistics of the monitors. 39

3.3 Monitor size characterization 39

4.1 BGP decision process 68

4.2 Summary of data collection 74

4.3 Statistics of data cleaning: avg number of removed traces per day for each type of anomalous traceroute. 76

4.4 Statistics of classification 77

4.5 Validation with BGP data for routing event identification and classification. 78

4.6 Statistics of root cause inference. 81

4.7 Event based validation: with a Tier-1 ISP's BGP data (0.29% of prefixes, 23 days). 86

4.8 Validation for two important clusters ($conf_{hP}$=30, $conf_s$=150) 87

4.9 Event based validation: with Abilene BGP data (6% of prefixes, 8 days). 88

5.1 Information commonly used to determine policies for differentiation. . 101

5.2 18 target ISPs: # of PoPs, # of PoP-PoP pairs, # of PoP-neighbor AS pairs. 112

5.3 K-S test results for content-based differentiation. 120

5.4 K-S test results for routing-based differentiation. 123

5.5 Customer vs. peer in previous-hop AS based differentiation. 126

5.6 An example of content-based differentiation confirmed using TOS (from planetlab1.arizona-gigapop.net to 193.58.13.1) 126

5.7 Applications ports used for TOS marking. 126

5.8 Network load effects for content-based differentiation: % of pairs with detected differentiation compared with using all samples. 129

5.9 Network load effects for previous-hop based differentiation: % of pairs with detected differentiation compared with using all samples. 130

6.1 Diversity of networks covered by our collected live IPs. 147

6.2 General statistics over the period of 11 weeks 152
6.3 Top 10 destination ASes experiencing most unreachable incidences. . 153
6.4 Forwarding loop incidences in the top 10 responsible ASes. 158
7.1 Data collection (All: all measured paths; Redundancy(3): paths traversed by at least 3 different sources). 183
7.2 Validation using BGP data for route instability and unreachability events. 184

CHAPTER I

Introduction

Despite the enormous popularity and success of the Internet leading to a wide range of network applications available today, we still cannot entirely depend on networks for time-critical wide-area applications, such as remote surgery, first-responder emergency coordination, and financial transactions. The main concerns to fully rely on Internet service is the occurrence of unexpected failures, attacks causing performance degradations [126]. Given the scale and complexity of the Internet, failures and performance problems can occur in different ISP networks, in different geographic locations, and at different layers, affecting the end-to-end performance [97, 103, 138]. It is thus important to quickly identify and proactively respond to potential problems which can be early symptoms of more serious performance degradation. In order to achieve this, network monitoring and diagnosis systems need to be set up to collect and analyze various types of data, which usually indicate network disruptions in different protocol layers and in different locations.

A large amount of research work has been proposed in the past on ISP-centric monitoring. Today's Internet service performance from an end-to-end perspective is determined by individual networks composing the network path, particularly Internet Service Providers (ISPs) of the core Internet network infrastructure, e.g., AT&T, and Sprint. Network performance data, e.g., latency and loss rate, are passively collected by setting up monitors inside their networks by each ISP individually [48, 125, 144]. These monitors are used to ensure the Service Level Agreements (SLAs) compliance offered to their customers. This monitoring approach is insufficient. Usually such SLAs are in the form of average values over monthly durations within each individual network. It is rather opaque in terms of how to evaluate the *user-perceived performance* on the end-to-end paths which traverse multiple ISPs. Moreover, given that most ISPs are reluctant about revealing details of their networks, they normally keep their routing and performance statistics publicly inaccessible. Therefore, all the above previous techniques cannot be easily used by end hosts who do not have any proprietary information. As a result, customers are in the dark about whether their service providers meet their service agreements. Similarly, ISPs have limited ways to find out whether the problems experienced by their customers are caused by their neighbors or some remote networks. They usually have to rely on phone calls or emails to perform troubleshooting [8].

The ability to monitor and pinpoint the network responsible for observed performance degradations is critical for network operators to quickly identify the cause of the problems and mitigate potential impact on customers. More importantly, enabling end-system based monitoring can significantly improve the Internet reliability and fairness. It accurately represents the application-perceived performance which directly relates to the Internet usability. Once the disruption is detected, the end-system can construct efficient recovery reactions to minimize the damage. End-system based monitoring can provide incentives for ISPs to enhance their service quality. It also enhances the capability for end hosts to more intelligently select ISPs and predictively reduce the impact of disruptions.

Motivated by the above observations, this thesis aims to design novel techniques to enable end-users to monitor wide-area network services, accurately diagnose the causes of observed network disruptions, and predicatively recover from any performance problems. This is an essential step towards improving accountability and fairness on the Internet, which can help customers assess the compliance of their service-level agreements (SLAs). Our approach differs markedly from recent work on routing and SLA monitoring in that it purely relies on probing launched from end-hosts and does not require any ISP proprietary information.

Building effective end-host based monitoring systems faces following key challenges. First, due to the limited CPU and network resources available at each end host, we need to aggressively reduce the probing overhead and monitor scalably. Second, given the limited view from each end host, it is difficult to locate the disruptions, not to mention diagnosing more fine-grained causes. Third, given end users often do not have much control over the network, it is challenging to mitigate the damage from end hosts.

Many research has been proposed to reduce the probing overhead [110, 136], or to improve the accuracy to locate the failure using correlation across hosts [67, 79]. These work mainly focus on coarse-grained diagnosis in terms of locating the failure in the data path for packet forwarding. The

My work focus on fine-grained cause monitoring and diagnosis on both *control plane* (routing data) and the Internet *data plane* (packet forwarding). It covers important research questions on three dimensions: effective and efficient ***monitoring***, accurate ***diagnosis***, and intelligent ***mitigation*** response, each of which is indispensable to building a complete Internet monitoring and diagnosis system. I take the approach of building large-scale, accurate and efficient network monitoring systems from purely end-hosts' perspective, which enables end hosts to diagnose and react to performance degradations in real-time. I elaborate the contributions in each dimension as follows.

Monitoring. The first step in building a monitoring system is to determine where to monitor. The first contribution in my works is that I carefully consider and demonstrate *the monitor selection which has great impact on the monitoring quality and the interpretation of the results.* Network monitors are the systems used to collect various performance data. A variety of networking research, e.g. troubleshooting, modeling, security analysis and attack prevention, all heavily depend on the monitoring results. Despite its importance, the monitor selection problem has not been well-studied in the past. My work studies the impact of diverse deployment schemes on answering diverse important research questions including Internet topology discovery, dynamic routing behavior detection, and inference of important network properties. Our study of route monitor selection provides insights on improving monitor placement. Our work is *the first* to critically examine the visibility constraints imposed by the deployment of route monitors on understanding the Internet [140].

Diagnosis. My second contribution is in designing novel techniques for accurate diagnosis the fine-grained causes of disruptions. The key principles of my approaches are to explore information across multiple protocol layers, and to diagnose across locations collaboratively. Previous work has studied monitoring in either control plane or data plane individually. Each approach has its own benefit: the former is less noisy and of less overhead and the latter can accurately capture the actual path and performance experienced by the application. I explore the benefit on both approaches to improve the accuracy and reduce monitoring overhead. Combining views from multiple locations can significantly improve the diagnosis accuracy.

More specifically, it is known that severe performance degradations can be caused by routing changes [103, 123, 138], which can last up to 30 minutes. It can also be the consequence of ISP policies, e.g. slowing down BitTorrent traffic [45]. Accordingly, I developed two systems to diagnose the disruptions induced by each of these two types of causes.

First, I focus on diagnosing the locations and root causes of routing-induced network disruptions [139]. The diagnosis component builds on the measurement results obtained. Correlating the traces from multiple locations, I design an inference algorithm to identify the minimum set of root causes that can explain most observations using a greedy algorithm. Our work is the first to enable end systems to scalably and accurately diagnose causes for routing events associated with large ISPs without requiring access to any proprietary data such as real-time routing feeds from many routers inside an ISP.

Secondly, I focus on detecting general performance difference induced by ISP intentionally differentiated treatment. This is motivated by the topic of "net neutrality" which has become a critical social and technical problem, as ISPs may use different ways to discriminate traffic, e.g., giving peer-to-peer application traffic lower priority, providing different qualities of service based on customer identities. I design novel application content-aware probing techniques to monitor the service provided for diverse applications and customers [137]. This helps detect traffic discrimination and ensures fairness of the Internet.

Mitigation. Ultimately, the monitoring and diagnosis results should be used for mitigating the damage or preventing being affected in the future. In the last two chapters of this thesis, I demonstrates two potential applications of end-host based monitoring systems.

The first application is short-term prevention of avoiding choosing the problematic route by exploring the predictability from history. In Chapter VI, we conduct a comprehensive characterization of many diverse routing changes using a measurement framework. The probing is triggered by locally observed routing updates. The probing target is an identified live IP address within the prefix associated with the routing change. We are able to detect most of the reachability problems caused by transient routing disruptions.

In the second application, I demonstrate its usefulness in providing information for long-term ISP evaluation by comparing ISPs using multiple metrics. I developed a system that monitors multiple metrics on multiple ISPs deployed on the Planetlab testbed. It can simultaneously perform both long-term and instantaneous real-time comparisons across multiple ISPs using key performance metrics such as loss and reachability. By correlating multiple metrics, we clearly demonstrate the necessity for comparing ISPs under different metrics to assist more informed provider selection or traffic engineering.

To summarize, it is my thesis that:

Using a novel collaborative and application-aware measurement methodology, end systems are capable of monitoring and diagnosing today's Internet accurately for the purposes of improving failure accountability and enhancing application performance.

This thesis work demonstrates the capability for end hosts to monitor and diagnose the Internet. It can be used in various applications, such as ISP selection and wide-area service placement.

This dissertation is structured as follows. I will first review the related work in network monitoring and diagnosis area in Chapter II. Next, I systematically study the impact of route monitor selection in Chapter III, which motivates the necessity for end-host based approaches. In Chapter IV, we will present the first component to diagnose routing disruptions. The system to detect traffic differentiation is described in Chapter V. Two applications of mitigations are shown in Chapter VI for short-term prevention of routing-induced problems, and in Chapter VII for long-term ISP evaluation. Finally, Chapter VIII concludes this thesis.

CHAPTER II

Background and Related Work

There has been many work on network monitoring and diagnosis in both the Internet and enterprise network, including studies on both control plane and data plane. This chapter describes about several research areas related to the work in the thesis. We first provide an overview of related work in the area of monitoring in the inter-domain routing plane. We then summarize the existing work of Internet measurement, monitoring, and diagnosis on the data plane. We finally provide a summary of the discussion of net-neutrality violations and its likely implementations. All these research is the foundation where this thesis is built upon and they inspire the new work proposed in this thesis.

2.1 Introduction to BGP

The Border Gateway Protocol (BGP) [105] is the de facto standard Internet interdomain routing protocol that Autonomous Systems (ASes) use to exchange information about how to reach destination address blocks (or *prefixes*). Each AS is a network entity with well-defined routing policies. BGP uses TCP (port 179) as its transport protocol, which provides reliable and in-order delivery. The base protocol is simple, leaving significant freedom for the network operators to specify various policies in path selection. In the following, we first describe the basic operation of BGP, the BGP dynamics affecting application performance, and the BGP security issues.

2.1.1 BGP basic operations

BGP sessions between routers within the same Autonomous System are *iBGP sessions* and can traverse through several IP hops. BGP sessions between routers belonging to different ASes are *eBGP sessions* and usually are established over a single hop to ensure low latency and loss.

There are four types of BGP messages: OPEN, KEEPALIVE, NOTIFICATION, and UPDATE. OPEN is used to establish the BGP session between two routers. Once the session is established, BGP neighbors send each other periodic KEEPALIVEs to confirm the liveness of the connection. If an error occurs during the life time of a BGP session, NOTIFICATION message is sent before the underlying TCP connection is closed. The UPDATE message is the primary message used to communicate information between BGP routers.

There are two types of BGP updates: *announcements* and *withdrawals*. An announcement of destination prefix P sent from router R_a in AS A to router R_b in AS B indicates that the path R_a is willing to carry traffic for R_b to destination P. Announcements indicate the availability of a new route to a IP prefix. It also indicates the routing decision of the advertiser. The announcement messages contain the length of the path as well as the actual path to reach to the destination. An announcement contains the following fields. *Neighbor IP* and *neighbor AS* indicates where the update is sent. *Origin* means that whether this route is generated internally (IGP) or externally (EGP). *Local preference* is a locally defined parameter to specify the preference in route selection process. Another internally-defined parameter *MED* is also used to specifying cold-potato routing policy. Finally, the *community* is used for prefix aggregation. Withdrawals indicate that the sender no longer has a route to the destination. It invalidates the last update message sent from the advertiser. It can be caused by network link failures, congestions, router upgrades or configuration changes.

BGP is a *path vector* protocol, as the AS_PATH attribute contains the sequence of ASes of the route. Each BGP update contains other path attributes such as NEXT_HOP, ORIGIN, MED (Multiple-Exit-Discriminator), ATOMIC_AGGREGATE, AGGREGATOR [105], *etc.*. All such attributes can influence the route selection decision. Some of the attributes such as ORIGIN, AS_PATH, and NEXT_HOP are mandatory. By representing the path at the AS level, BGP hides the details of the topology and routing inside each network. BGP is *incremental*, i.e.every BGP update message indicates a routing change.

In addition, BGP is *policy-oriented*. Rather than selecting the route with the shortest AS path, routers can apply complex policies to influence the selection of the *best* route for each prefix and to decide whether to propagate this route to its neighbors. To select a single best route for each prefix, a router applies the decision process [105] to compare the routes learned from BGP neighbors. The flexibility in defining routing policies makes BGP more difficult to understand. Moreover, the complexity of the flexibility in expressing BGP policies continuously grows. For instance, an AS can make routing decision based on its commercial relationship. There can be dual transit/peer relationship or commercial relationship on a per-destination base, which makes BGP protocol more complicated. Finally, BGP is a stateful protocol, meaning that only the changes of the state will be exchanged but no periodic refreshment. The stateful design enables BGP's capability to scale to the entire Internet.

2.1.2 BGP dynamics

Local BGP changes in a single AS may propagate globally and result in routing changes in all the ASes on the Internet. In other words, local instability may result in global performance. To select a single best route for each prefix, a router applies the decision process [105] to compare the routes learned from BGP neighbors. First, a route is ignored if the next hop is unreachable. Then the BGP speaker will select the route with the highest local preference value, which is often set according to the economic relationship between ASes. The best practice to set the local preference is that customers route is preferred over providers routes, which is preferred over peer routes. Intuitively, provider provides transit services to all its customers while peers are only responsible for carrying traffic to their own customers. The relationship can be very complex, which is expressed in the local preference field in the update for each destination. Next, the BGP speaker will choose the route with shortest

AS path, which indicates better path performance. Third, the router will prefer the route with lowest MED attribute to implement cold-potato routing with neighboring networks. Then it prefers EBGP route over IBGP route in order to implement hot-potato routing. If all the steps above are the same, then it select one with shorter intra-domain distance from IGP protocols, e.g.OSPF. Finally, it breaks the tie based on router ID.

Although the process of choosing the best route is clear, there is no network-wide protocol support to express the routing policies and ensure its correctness. There has been work proposing to ease the job of configuring networks automatically. eamster *et al.* [53] applied statical analysis to find faults in BGP configurations. Caesar *et al.* [34] proposed a centralized routing control platform (RCP) to facilitate configuration and route selection inside an AS. Karlin *et al.* [64] proposed an enhancement to BGP to slow the propagation of anomalous routes, similar to our design. Karpilovsky and Rexford [66] recently proposed an algorithm to reduce router memory usage by discarding alternate routes and refreshing on demand.

Another serious problem in BGP performance is that local routing decision may lead to BGP oscillation that multiple BGP speakers continuously change their route selection and never reach a stable state. Such worst-case routing convergence has been proved to exist and its detection is NP-hard. Routing changes on the Internet are mostly caused by failures or configuration changes. They occur quite frequently. At the interdomain level, one can easily observe more than 10 updates per second to a wide range of destinations from a large tier-1 ISP such as Sprint using publicly available BGP data from RouteViews [13].

Moreover, routing dynamics directly influence the data plane, i.e.the packet forwarding behavior. Previous measurement studies [73, 85, 123] have already shown that routing changes can cause transient disruption to the data plane in the form of packet loss, increased delay, and forwarding loops. Data plane failures are often caused by inconsistent forwarding information of routers involved in routing changes [123]. During routing convergence, some routers may lose their routes [122] or have invalid routes [73]. Routing policies, timer configurations, and network topologies are just some of the contributing factors [122, 123]. For instance, transient loops can be caused by temporarily inconsistent views among routers. Persistent loops are more likely due to misconfigurations [127].

2.1.3 BGP security issues

The Internet originated from a research network where network entities are assumed to be *well-behaved*. The original Internet design addresses physical failures well, but fails to address problems resulting from misbehavior and misconfigurations. Routers can misbehave due to misconfigurations [82], impacting network reachability. Today, the Internet has no robust defense mechanisms against misbehaving routers, leaving the routing infrastructure largely unprotected [91]. One of the most widely known and serious misconfiguration occurred in 1997, when a customer router at a small edge network by mistake advertised a short path to many destinations, resulting in a massive blackhole disconnecting a significant portion of the Internet [28]. This example illustrates the need for an easily deployable protection mechanism to prevent local forwarding decisions from being polluted. It has been well-known that the Internet routing is vulnerable to various misconfigurations and attacks [32]. Recent studies [82, 116, 141] have focused on identifying routing anomalies using BGP data.

Given the lack of security in today's routing protocols, both the research and the network operator community have already proposed secure solutions. Several protocols have been proposed to enhance BGP security by incorporating cryptographic mechanisms to provide confidentiality, integrity, and origin authentication. S-BGP [69] is the first comprehensive secure routing protocol. It relies on two public key infrastructures (PKIs) to secure AS identity and association between networks and ASes. Each route contains two attestations (digitally signed signatures), one for the origin authentication and one for the route integrity. In reality, due to the large number of sign and verify operations, S-BGP is too costly to deploy. For example, SPV [62] utilizes purely symmetric cryptographic primitives, a set of one-time signatures, to improve efficiency. Butler *et al.* [33] reduced the complexity of S-BGP by exploring path stability. Along the angle of reducing the overhead of asymmetric key, Hen *et al.* [130] proposed a scheme using key chain to improve its performance. Only one existing work, Secure Origin BGP (soBGP) [93], focuses on providing address attestation. However, soBGP still uses one PKI to authenticate the address ownership and AS identity. Each soBGP router first builds a topology database securely, including the address ownership, organization relationship and topology. Aiello *et al.* builds an address ownership proof system [16] which still uses a centralized infrastructure requiring gathering address delegation information.

However, we have witnessed a rather slow deployment. Furthermore, most of them do not eliminate the possibility of router misconfigurations and their associated impact. Complementary approaches exist without modifying BGP to identify configuration errors. IRV [57] defines such a service to mitigate malicious or faulty routing information by relying on collaboration among several networks. Feamster *et al.* [53] applied statical analysis to find faults in BGP configurations. Caesar *et al.* [34] proposed a centralized routing control platform (RCP) to facilitate configuration and route selection inside an AS. Karlin *et al.* [64] proposed an enhancement to BGP to slow the propagation of anomalous routes, similar to our design. Karpilovsky and Rexford [66] recently proposed an algorithm to reduce router memory usage by discarding alternate routes and refreshing on demand.

2.2 Characterization of Internet control plane

There has been a significant amount of work on characterizing Internet control plane behavior, including passive topology discovery, understanding BGP dynamics, and root cause analysis. Many related work relies on analyzing BGP updates collected passively. There are two types of BGP data, the table dumps and the routing updates. The former keeps of the snapshot of the best routes used by the router contributing the data. The latter contains a sequence of routing updates received over time. The other type of measurement relies on injecting controlled routes to the unused prefixes into the Internet.

There exist several public route monitoring systems, such as Route Views [14] and RIPE [10], which have been deployed to help understand and monitor the Internet routing system. These monitoring systems operate by gathering real-time BGP updates and periodic BGP table snapshots from various ISP backbones and network locations on the Internet to discover dynamic changes of the global Internet routing system. The routers in the ISP networks exposes default-free routing data to these collectors with real-time BGP session. Various research studies have been conducted relying on these data, including network topology discovery [59], AS relationship inference [24, 43, 44, 56, 115], AS-level path prediction [86, 90], BGP root cause analysis [54], and several routing anomaly detection schemes. Most of these studies process the routing updates and the BGP table snapshots from the route monitoring system, extracting information such as AS-level paths and their changes over time, to study the dynamic routing behavior.

2.2.1 Network topology discovery

BGP data is an important information source for understanding the Internet topology. Very basic network properties are critical for understanding the Internet routing system. These properties include AS connectivities, IP prefix to origin AS mappings, stub AS information and stub AS's provider information, multi-homed ASes, and AS path information. A large group of related work focus on examining the graph properties of topology snapshots such as node degree distribution [50, 107, 117]. The theoretical analysis shows that the Internet exhibits power law distribution. The seminal work by Faloutsos*et al.* [50] found that the degree of inter-domain AS level topology exhibits a power-law degree distribution. Stickily speaking, the degree distribution of the AS topology does not conform to a strict power law but with a heavy tail. Later work seeks for explanation for the highly-variable degree distribution. One possible explanation is that the AS size determines the AS topology, but AS size varies significantly due to business reasons.

Many of these work on Internet topology are using BGP snapshots or traceroutes over longer time period [96]. The snapshot-based approach may miss significant number of links due to the limited visibility induced by vantage points. Such analysis are limited by the vantage points of BGP data. Besides static topology snapshots, other work explores the evolution of the Internet over time. The most challenge problem is to identify the real topology changes from the observed topology. A most recent work studies the topology evolution as a liveness problem consisting of consistent-rate birth process and death process. It points out that the impact of transient routing dynamics on topology characteristics decreases over time. It subsequently proposes a model that predicts the real topology given the observed topology with a confidence interval.

BGP allows each AS to choose its own policy in choosing the best route. One of the important factors determining the route selection policy is the commercial relationships between autonomous systems. It is essential to infer the AS relationship in order to fully understand the Internet topology and to predict AS-level routing path. The relationship between ASes can be classified as customer-provider, peering, and sibling relationship. Such relationship is usually not available to the public due to privacy concerns. The seminal work by Gao proposes an algorithm to infer the AS relationship based on routing paths from BGP tables [56]. From the BGP table, we can easily construct a AS-level connected graph. The algorithm is based on the heuristic that the size of the AS is proportional to its node degree in the graph. The relationship between ASes are then built based on the size of the ASes. Other work subsequently proposes methodologies to improve the accuracy using registry data or other data sources [25, 43, 44, 115]. Knowing commercial relationships among ASes reveals network structure and is important for inferring AS paths.

2.2.2 BGP diagnosis and root cause analysis

T. Griffin [31] observed that "In practice, BGP updates are perplexing and interpretation is very difficult." Given routing instabilities occur rather frequently and complex BGP policy-based routing, understanding the root causes of various updates is challenging. Several research has been proposed to locate the autonomous system responsible when a routing change occurs by correlating the routing updates for many prefixes. These techniques have proven to be effective in pinpointing routing events across multiple ISPs. Feldmann *et al.* [54] proposes an algorithm based on the intuition that the responsible AS should appear either on the old path or the new path, given an AS-level path change. Then they use simulation approach to inject failures on the real AS topology from BGP updates. The methodology is verified using measurement data with several heuristics.

Different from the previous work of diagnosing BGP from multiple ASes, the work by Wu *et al.* [125] focuses on diagnosing BGP updates within a single ISP. It is also the closest related work on identifying routing disruptions to Chapter IV.

However, even within a single ISP, the amount of BGP updates may prevent the operator from pinpointing the key problems. This work aims at summarizing the large volume of updates to a small number of reports that highlights the significant BGP routing changes impacting large amount of traffic. Their system groups all BGP updates from all the border routers of the ISP to several events with similar changing patterns occurring close in time. They combine with intra-domain OSPF data as well as router log to validate the system.

A follow-up work by Huang *et al.* [63] performs multivariate analysis using BGP data from all routers within a large network combined with router configurations to diagnose network disruptions. In contrast, we do not rely on such proprietary BGP data, and we can apply our system to diagnose routing changes for multiple networks. There are also several projects on identifying the location and causes of routing changes by analyzing BGP data from multiple ASes [54, 119]. However, it is difficult to have complete visibility due to limited number of BGP monitors. Note that our system is not restricted by the deployment of route monitors and can thus be widely deployed.

2.3 Characterization of Internet data plane

ISP AS-level topology can be inferred from the BGP data, which can be accessible from public route repository. However, not all the ASes are willing to share the BGP data due to overhead and privacy concerns. In the data plane, the router-level Internet topology is even more difficult to obtain from the research community.

Without cooperation from the ISPs, researchers have been seeking approaches to characterize the data-plane performance on the Internet. In the following section, we first discuss several performance metrics for data-plane performance characterization and methodologies to measure these metrics. Then we discuss existing topology discovery work on data plane.

2.3.1 Data-plane measurement methodology

Traceroute is a tool widely used for path discovery on the Internet from the end hosts. It reports the IP addresses for the routers along the path from a source to a destination. The source host sends packets with limited Time-to-live (TTL) values in the IP header. Routers along the path decrement the TTL value by one each hop. When the TTL value is zero, the router discards the packet and sends back an ICMP TTL expired message. The sender discovers routers along the path using the source IPs of the ICMP messages.

The traditional traceroute faces several measurement inaccuracies due to topology or network device artifacts. For instance, multiple equal cost path can exist between a pair of ingress and egress routers in the intra-domain for load-balancing purpose. Routers can spread their traffic across multiple equal-cost paths using per-packet, per-flow, or per-destination policies. The multi-path existence means that there is no longer a single path from a source to a destination. In packet-based load-balancing, one packet may traverse any of the possible paths. For per-flow based balancing, different flows between a source/destination pair may go through different paths. Thus, traditional traceroute cannot discover the true nodes and links traversed. Augustin*et al.* [19] introduces Paris-traceroute, which controls the probe packet header fields to force all packets follow the same path to a destination. It can distinguish per-flow based load-balancing from the per-packet based load-balancing.

Traceroute reports the routers along the paths using the source address of the "Time exceeded" ICMP messages. However, the address is actually the outgoing interface of the router traversed. It causes the alias problem in inferring the network topology, i.e.determining which IP address belongs to which router. Otherwise, the inferred topology may contain more routers than link than the actual topology. Existing work solves the alias resolution problem by using various techniques, including mapping the IP addresses to DNS names, combining addresses with similar reverse TTL value [135].

Besides path discovery to gain a basic understanding of the Internet topology, researchers are also interested in metrics that directly affects the application, i.e.latency, loss rate and bandwidth. Traceroute and ping are the two common tools widely used for measuring network latency between a source/destination pair or from a source to an intermediate router.The most popular tool for packet loss rate measurement is ping. Ping sends ICMP echo packets to the target host with fix intervals. Loss rate is computed by the number of response packet not received from the target host. Most work use the discrete sampling approach with fixed-interval probe packets to measure loss rate. To distinguish the loss rate on the forward path versus the reverse-path, it is commonly using the large packet as the probe packet on the forward path, as large packets are more likely dropped than the small ICMP replies [81]. Sommers*et* *al.* [109] showed that the discrete sampling approach cannot capture the true loss rate due to the bursty nature of loss behavior in the network. Their experiments show that even for probes with Poisson intervals are not sufficient. This paper proposes Badabing, a tool that explores the characteristics of loss episode frequency and duration to capture loss rate more accurately with low overhead.

Another important metric to quantify network service quality is the available bandwidth of the bottleneck links. It is challenging to accurately infer the available bandwidth without load information from all relevant links. Most of the tools rely on two-ended control by sending large number of packets to cause congestion on the target link. Common techniques use a chain of packets with different size and measure the latency difference of various-sized packets [62]. By customizing the TTL value in the packet header, these approaches exploits how different-sized packets are handled by the routers. The bottleneck location can be inferred via hop-by-hop measurement.

2.3.2 Topology discovery

ISPs do not reveal their router-level topology to the research community as it is often considered confidential. Thus, researchers usually rely on end hosts to infer router-level topology. Understanding the router-level topology can shed light on whether the synthetic topology for simulation is representable. Spring*et al.* [110] presents Rocketfuel, a light-weight system, to infer accurate ISP router-level topology. To achieve low measurement overhead, Rocketfuel uses routing information to select a set of paths that are likely traversing the ISPs under study. To further reduce overhead, it suppresses the probes that are likely to traverse redundant paths following the same segment inside the ISP network. Traceroute-based approach may not be very accurate due to alias problem, i.e.one router may return different IPs depending on different interfaces traversed. This work produces detailed maps for ten large ISPs, which is an important base of the work in this thesis.

2.3.3 Network tomography

Network Performance Tomography is a technique that correlates performance measurements across multiple end-to-end paths infer the performance of the internal links. Using statistical inference, it can infer packet loss, delay, and even the underlying topology, from purely end-to-end probes [47]. It is first applied on identifying the packet loss rate of individual link in a multicast infrastructure. To improve the accuracy and practical aspects, they emulate multicast probes with sets of unicast packets and study simple performance level correlations among dedicated packet streams. Bu*et al.* [30] infers loss rate of individual link from end-to-end multicast measurement in a collection of trees. They propose the minimum variance weighted average (MVWA) algorithm infers the loss rate on each tree separately and aggregate the inference across trees using the expectation-maximization algorithm. The algorithm can achieve reasonable accuracy even with certain portion of measurement missing given its statistical nature. The method can also be easily extended to latency metrics.

Besides network tomography in multicast networks, Zhao*et al.* [142] applies the tomography concept on the Internet. The accuracy of the tomography approach may not be high on the Internet given various sources of measurement noise. Instead of identifying the exact link causing congestion, it proposes a minimal identifiable link sequence algorithm to pinpoint the minimal length of link sequence with undistinguishable performance properties.

Overall, tomography approach uses signal processing and statistical techniques to infer link level property or shared congestion based on end-to-end measurement in both IP networks or multicast networks. The accuracy of tomography approaches is subject to uncertainty from the statistical assumption of the network. Moreover, the algorithm can be fundamentally under-constrained as there exist unidentifiable links that cannot be easily distinguished. It also requires a large amount of end-to-end measurement to achieve high accuracy.

2.4 Interaction between control and data plane

A significant number of measurement studies have been conducted to examine the impact of routing changes on data plane performance degradation [15, 52, 74, 77, 100, 122, 123]. With large volume of routing updates on the Internet, it is essential to understand how routing changes result in end-to-end performance degradation. Moreover, understanding how topology, configurations and routing policies affect the data-plane performance can help improve the network performance in the long term. Wang*et al.* [123] studies the end-to-end performance on the Internet under controlled routing changes. They examine delay, loss rate, jitter, and out-of-order packets using probes from planetlab to the destination with scheduled routing changes. The paper found that routing changes can lead to severe packet loss rate as long as 20 seconds. the iBGP configuration is the major cause of the performance degradation during routing convergence. This thesis also examines routing change's impact on data plane but under the metric of reachability.

Other work focuses on examining other aspects of routing changes' impact. Labovitz*et al.* [74] presents a two-year study of Internet routing convergence. It focused on the stability of the path between two ISPs by artificially injecting routing failures. It shows that the path restoral delay can be significant due to unexpected interaction of router times and specific router vendor implementation. The duration and location of end-to-end path failures are studied and correlated with BGP routing instability in [52]. Data plane transient failures are also widely studied in [60, 98, 100, 113, 122].

It has been shown that transient loops can be caused by inconsistent or incomplete views among routers during routing convergence [143], while persistent loops are more likely a result of misconfiguration and can be explored to create flooding attacks [127]. In [20], light-weight data plane countermeasures are used to detect routing protocol and data plane attacks, which can be used in the routing architecture. Our work uses a wide range of measurements in analyzing the impact of the data plane failures triggered by routing updates. This thesis focuses on exploring the coarse-grained performance degradation in terms of reachability caused by routing changes. We measure the data plane performance via active probing triggered by routing updates.

2.5 Infrastructure-based monitoring and diagnosis

The more closely related work is measuring the ISP performance. The ISPs set up measurement platform to monitor its own network latency, loss rate, reachability, and bandwidth. In this section, we will first discuss several work on SLA monitoring methodologies. Then we summarize existing work to diagnose network disruptions from ISPs' perspective.

2.5.1 SLA monitoring

The Service Level Agreement (SLA) is a set of performance metrics that an Internet Service Provider (ISP) needs to ensure for its customers. It contains various metrics such as average and maximum packet loss rate, delay, jitter, and network reachability. Each ISP has its own promised performance in the agreement with its customers, namely, the Service Level Agreements(SLAs). Accurate and efficient monitoring of SLA compliance is critical for an ISP as it directly impacts the ISP's revenue and reputation. The ability to monitor SLA compliance is essential for both customers and ISPs to optimize their network performance. Many SLA compliance monitoring techniques have been proposed in previous work. Today's SLA specifications contain metrics such as average and maximum packet loss rate, delay, jitter, and network reachability [3, 7, 9, 12].

Many previous work on SLA monitoring is from a single ISP's perspective. ISPs set up monitors to collect routing data from all the routers within its network. For instance, the design and deployment of the OSPF server for a single ISP is studied to provide real-time and accurate view of the topology of an IP network and the path traversed by different flows. It captures the dynamic changes of the network in face of link or router interface failures. It can also help monitor the effect of routine maintenance and configuration changes.

A recent work [48] uses network performance tomography to monitor SLA inside an ISP with relatively low measurement overhead. This work presents a methodology to use a single multi-objective probe stream to evaluate loss rate, delay, and delay variation. A centralized probe scheduler interacts with the senders to make use one probe for multiple purpose. It also proposes a methodology to quantify the lower bounds on all the path performance metrics using measurements from a subset of paths. In contrast, this thesis focuses on monitoring SLA compliance from end-system's perspective.

There exists commercial third-party systems to help ISP monitor performance such as the commercial KeyNote system [70]. KeyNote system measures the latency and loss rate for paths internal to the ISPs and paths between a pair of ISPs. It place measurement servers co-located with the ISP point of presence (PoPs) and measure the paths between them. This approach requires cooperation from the ISPs thus limits its deployment scope. The loss rate measurement is based on HTTP web downloads which heavily depends on TCP congestion control and is coarse-grained.

Overall, the SLA monitoring focuses only on paths within an ISP, which does not capture any end-to-end performance to the destinations. Since Internet routing is destination based, the path to reach the destination may not be the same as the path to another measurement node.

2.5.2 Network diagnosis from ISPs' perspective

The ability to efficiently and accurately troubleshoot network disruptions is of critical importance to Internet Service Providers (ISPs). In most ISP networks, network events occur continuously in different locations and at different layers. It is thus important to quickly identify and proactively respond to potential problems which can be early symptoms of more serious trouble. In order to achieve this, network monitors have been set up to collect various types of data, which usually represent network disruptions in different layers. To identify important network disruptions from such huge amounts and various types of data, it is necessary to identify the correlation across different network changes,which can further help investigate and locate the disruption.

Traditional troubleshooting in an ISP network usually requires time-consuming manual data correlation by operators despite the availability of numerous tools generating alerts using one or a small number of data sources. However, troubleshooting in a network consists of diverse protocols and elements. The current approach has several limitations. If each data source is analyzed and acted upon independently, the resulting mitigation response may be suboptimal and redundant. Furthermore, such an approach cannot easily identify the effect and possible cause of a problem. For example, purely analyzing BGP flapping routes alone cannot always identify whether it is caused by peering link failure, internal disruption, or customer network instability. Finally, it cannot identify longer-term correlations, which is in fact less sensitive to correlation errors. To identify shared risks across network elements, Kompella *et* *al.* [72] proposes a system called SCOPE (spatial correlation engine) to automatically identify likely root causes across layers. It models the network as several risk groups and applies it to localize the fault across IP and optical network layers. Later work proposes a framework to identify hidden correlations across multiple data sources and across geographic locations [84]. The correlation information can assist identifying root cause and performing efficient mitigation strategies.

2.6 End-host based monitoring and diagnosis

Besides the above mentioned ISP-centric approaches, there has been a large body of work on enabling end-user to measure and diagnose the Internet. It is challenging to effectively measure and compare the quality of network services offered by different core ISPs from end hosts. When disruptions occur, the capability to quickly locate the failure and to proactively bypass is important to improve the reliability of end-to-end paths. Much work has been proposed to use end-host based probing to identify various network properties. For example, Rocketfuel [110] discovers ISP topologies by launching traceroute from a set of hosts in an intelligent manner to ensure scalability and coverage. iPlane [79] estimates the Internet path performance using traceroutes and prediction techniques. There have been many other research measurement infrastructures [39, 42, 55, 58, 94] for measuring network distance with performance metrics such as latency and bandwidth. Another example is PlanetSeer [136] which uses active probes to identify performance anomalies for distributed applications. The key difference from these measurement efforts is that our work focuses on using collaborative traceroute probes to diagnose network problems, in both routing and performance aspects, associated with large networks. The closest work on monitoring ISP performance using end-host based probing is NetDiff [83] which measures latency for one ISP at a time. Our work differs in that our approach efficiently monitors traffic discrimination for multiple ISPs simultaneously to enable cross ISP comparison.

After reviewing the large body of related work of network monitoring for general purpose, next, we discuss the related work to discover a specific types of performance issue, the net-neutrality violations. Network neutrality is a topic that has recently drawn significant attention [27, 41, 128, 129]. Crowcroft has pointed out that network neutrality has different technical definitions and feasibility in various types of network models [41]. From the detection perspective, a recent study uses end-host based active measurement to identify one type of discrimination: port blocking [27]. For prevention, new measurement techniques [21, 22] based on encryption and multi-path routing have been proposed to conteract ISPs' potential selective treatment and discrimination of measurement traffic. Our system focus on systematically detecting various types of discrimination on the Internet. Finally, our work also resembles a broad class of measurement studies to reverse engineer the Internet [80, 111].

To summarize, past work in the area of Internet monitoring and diagnosis has demonstrated the limitation of ISP-centric approaches and illustrated the potential opportunities for enabling more accurate end-host based monitoring systems. Next, we turn to discuss the unique approaches in our research to address various problems under this context.

CHAPTER III

On the Impact of Route Monitor Selection

3.1 Introduction

There exist several public route monitoring systems, such as Route Views [14] and RIPE [10], which have been deployed to help understand and monitor the Internet routing system. These monitoring systems operate by gathering real-time BGP updates and periodic BGP table snapshots from various ISP backbones and network locations to discover dynamic changes of the global Internet routing system. Various research studies have been conducted relying on these data, including network topology discovery [59], AS relationship inference [24, 43, 44, 56, 115], AS-level path prediction [86, 90], BGP root cause analysis [54], and several routing anomaly detection schemes. Most of them process the routing updates from the route monitoring system to study the dynamic routing behavior.

These studies relying on BGP routing data usually assume that data from the route monitoring systems is reasonably representative of the global Internet. However, no existing work has studied the limitations of route monitoring systems and the visibility constraint of different deployment scenarios. For example, recent work using these data to detect malicious routing activities, such as address hijacking [61, 65, 75, 76] could potentially suffer from evasion attacks similar to those affecting traffic monitoring systems [106]. The accuracy of such anomaly detection schemes depend heavily on the coverage of the route monitoring system. The limitation of the route monitor system is critical for any system relying on BGP data from multiple vantage points.

It is usually impossible to obtain routing data in real time from every network due to the scalability issue and privacy concern. Obtaining one feed from one AS often provides a restricted view given there are many routers in an AS, each with a potentially different view of routing dynamics. Additional BGP feeds are useful for detecting routing anomalies, traffic engineering, topology discovery and other applications. But adding an additional feed usually requires interacting with a particular ISP to set up the monitoring session. Therefore, an urgent question is to understand the generality and representativeness of the given monitor system, and to understand how to select monitor locations to maximize the overall effectiveness of the route monitoring system.

Some existing work [23, 104, 133] studied the limitation of existing monitor placement and monitor placement algorithms [95] in terms of topology discovery. In this work, we study the impact of monitor network location constraints on various research work in the Internet routing community. We are the first to examine the visibility constraints imposed by the deployment of route monitors, impacting a diverse set of applications. To understand the difference among current deployment settings, we analyze three deployment scenarios: all Tier-1 ISPs only, Route Views and RIPE setup, and a setup combining many public and private vantage points. We further study four simple schemes of network monitor selection and the resulting impact on multiple metrics based on the applications using the data. Our analysis shows that current public monitors already provide good coverage in various applications we study.

The remaining of this chapter is organized as follows. In Section 3.2 we introduce the methodology of our study, followed by a short discussion comparing three deployment scenarios in Section 3.3. We study in detail several different monitor selection schemes in Section 3.4.

3.2 Methodology

In this section, we describe our methodology, including the data and various metrics for comparing monitor selection schemes motivated by several common but important applications.

3.2.1 Route monitor locations

The BGP data we used in our study are collected from around 1000 monitoring feeds, including public data sources such as Route Views [14] and RIPE [10], feeds from the local ISP, and data from private peering sessions with many other networks, covering more than 200 distinct ASes, which are not in the public feeds. In the remainder of the chapter, we use the term *monitoring feed* to refer to a BGP data source from a particular router. We define a *vantage point* to be a distinct AS from which we collect BGP data from. Note that feeds from different routers in the same AS may provide different information, and we leave the study of the difference between feeds in the same AS for future work. We use one monitoring feed from one vantage point. For ease of comparison across vantage points, we only choose feeds with default-free routing tables (with entries for all prefixes), and create a data set called *LargeSet* consisting of data from 156 ASes for our subsequent analysis.

The BGP updates are collected from a set of route monitors, each of which establishes peering session with one router in each network being monitored. Note that our study is inherently limited by the BGP data we have access to and we attempt to draw general conclusions independent of the data limitation. Although the BGP data from all available monitors is still not the ground truth for the whole network, we study different applications using data from different sampling strategies and compare with this *LargeSet*. Developing more intelligent monitor placement algorithms is part of future work.

To understand static network properties, instead of using a single table snapshot from each feed, we combine multiple snapshots taken at different times with routing updates from each feed whenever available. This helps improve the topology completeness as many backup links are only observable during transient routing changes. We use two snapshots of tables from each monitoring feed including feeds from about 100 ASes, along with six months of updates and tables from Route Views, RIPE and a local ISP from May 2006 to Oct. 2006. The resulting network topology contains 25,876 nodes(ASes) and 71,941 links. We list the properties of current peers that

Route Views and RIPE have in Table 3.2.

To compare different deployment strategies, we construct three sets of realistic deployment scenarios. First, to understand the visibility of the core of the Internet, we select only 9 well-known Tier-1 ISPs to be monitors, including AS numbers: 1239, 174, 209, 2914, 3356, 3549, 3561, 701, and 7018. Second, we use only feeds from commonly used Route Views and RIPE. Third, we use LargeSet to obtain the most complete topology from all available data. We denote the three deployment scenarios as Tier-1, Route Views, and LargeSet, respectively.

We focus on three types of applications relying on BGP data, namely (1) discovery of relatively stable Internet properties such as the AS topology and prefix to origin AS mappings, (2) discovery of dynamic routing behavior such as IP prefix hijack attacks and routing instability, and (3) inference of important network properties such as AS relationships and AS-level paths. Note that the first two applications simply extract properties directly from the routing data. The performance of the third one depends not only on the data but also the algorithm used for inference. We describe these applications in more detail below.

3.2.2 Discovery of static network properties

BGP data is an important information source for understanding the Internet topology. Very basic network properties are critical for understanding the Internet routing system. These properties include AS connectivity, IP prefix to origin AS mappings, identifying stub AS information and its provider's information, multi-homed ASes, and AS path information. Intuitively, including vantage points from the core is more beneficial as a larger number of network paths traverse the core networks. Previous work [35, 59, 134] has shown the influence of data sources besides BGP table data, e.g.traceroute data and routing registries, on the completeness of inferred AS topology. We extend this analysis to two other properties: (1) multihomed stub ASes to understand edge network resilience and potentially increased churn in updates, and (2) AS paths, which are difficult to infer.

3.2.3 Discovery of dynamic network properties

Dynamic properties of the routing system are of strong interest for studying routing instabilities, e.g.due to misconfigurations, and detecting anomalies. Understanding such properties is useful for troubleshooting and identifying possible mitigation to improve routing performance. We focus on two representative applications here: monitoring routing instability and IP prefix hijack attack detection.

Routing instability monitoring: Routing updates are a result of routing decision changes in some networks caused by events such as configuration modifications, network failures, and dynamic traffic engineering. Comprehensively capturing Internet routing changes is useful for important applications like troubleshooting, routing health monitoring, and improved path selection.

IP prefix hijacking detection: One of the original goals of the public route monitoring systems in Route Views and RIPE is troubleshooting. Nowadays they are increasingly used for the timely detection of malicious routing activities such as prefix hijacking attacks. Current hijack detection systems in control plane [61, 75] rely on detecting inconsistency in observed BGP updates across vantage points. However, the detection system may not detect all attacks due to limited visibility. In this work, we study the impact of different monitoring deployment setups on the detection coverage.

Intuitively, an attack is missed if no vantage point of the monitoring system adopts the malicious route. Thus, we define attack evasion as follows. For a monitoring system $SM = m_1, m_2, \ldots, m_n$ with n monitors, given an attacker A, a victim V, and the hijacked prefix p, if $\forall i$, $Pref^{A}_{m_i}(p) < Pref^{V}_{m_i}(p)$, where $Pref^{A}_{m_i}(p)$ is the route preference value for p announced from A as observed by m_i, then attacker A can hijack V's p without being detected.

3.2.4 Inference of network properties

The third class of application studied relates to properties inferred from the above basic properties from BGP data.

AS relationship inference: There is much work [24, 43, 44, 56, 115] on inferring AS relationships from BGP AS paths. Knowing commercial relationships among ASes reveals network structure and is important for inferring AS paths. In this work, we study the commonly-used, Gao's degree-based relationship inference algorithm [56].

AS-level path prediction: Accurately predicting AS paths is important for applications such as network provisioning. In this work, we compare two path prediction algorithms under various monitor deployment settings. We use the recent algorithm [86]

Category	Tier-1	Route Views	LargeSet
Number of ASes	25732	25801	25876
Number of AS links	51223	56000	71941
Profit-driven prediction	34%	39%	43%
Length-based prediction	67%	76%	73%

Table 3.1: Comparison among three deployment scenarios.

which makes use of the inferred AS relationships, and study both profit-driven and shortest-path-based route selection. For the profit-driven policy, the route selection prefers customer routes to peering routes and over to provider routes. Note that predicted paths for both approaches need to conform to relationship constraints [56]. We also study the recent work [89] which does not use AS relationships but instead exactly matches observed paths.

To improve scalability, we eliminate *stub AS nodes*, or customer ASes that do not provide any transit to other ASes. The graph without stub nodes contains only 4426 (16% of all nodes) and 25849 links (15% of all links). For completeness, we also simulate the path prediction to 50 randomly sampled stub ASes. We include these 50 stub ASes and their links.

3.3 Deployment scenario analysis

We first analyze the three deployment scenarios, Tier-1, Ruter Views, and LargeSet defined in Section 3.2.1. We study the impact of these three settings on applications of AS topology discovery, AS relationship inference, and AS-level path prediction.

Data	Tier				Geographic location			
	1	2	3	4	Europe	Asia	Africa	America
Route Views	9	40	58	12	37	4	1	77
LargeSet	9	82	60	5	46	4	1	105

Table 3.2: Statistics of the monitors.

Data	Address			Degree			Customer		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Route Views	156	65313	1561473	3	247	2922	0	112	2899
LargeSet	156	116989	1561473	1	344	2922	0	177	2899

Table 3.3: Monitor size characterization

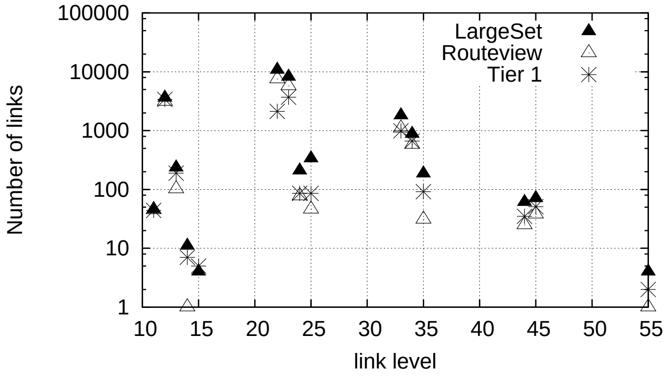


Figure 3.1: Distribution of observed links across tiers.

Table 3.1 summarizes the comparison across the three setups. Confirming previous studies [23, 133], we find that the largest monitor set, LargeSet, observes much more links but only slightly more non-private ASes. The additional ASes in the LargeSet are mostly at the edge. Using Gao's degree-based relationship inference algorithm, we compare the accuracy of inferred paths comparing with paths in BGP data in terms of path length. Note that the improvement is small for path prediction with increasing vantage points. Interestingly, using the largest data set lowers the length-based prediction accuracy. These results imply that Gao's algorithm is reasonably stable with changes in the BGP data.

We list the network properties of current peers of both Route Views and LargeSet in Table 3.2. We use the tier definition specified in previous work [115]: Tier-1 means closest to the core Internet and Tier-5 is associated with stub or pure customer ASes. We also analyze each AS in the aspects of geographic location. Next, we study the number of IP addresses it announces, its degree and its customers in Table 3.3. The additional ASes in LargeSet are mainly Tier-2 ASes in US, with large number of addresses and degree.

To understand which links are identified using a larger data set, we plot in Figure 3.1 the topological location of links in each data set. The X-axis indicates the link level, defined by the tier value of the two ASes associated with the link sorted in increasing order. For example, there are 10 links observed from LargeSet between nodes in tier-1 and tier-4 at the X value of 14. The hierarchy level for each node is assigned according to the relationship inferred using all the data available. As expected, the additional benefit of observed links are mostly at the edge.

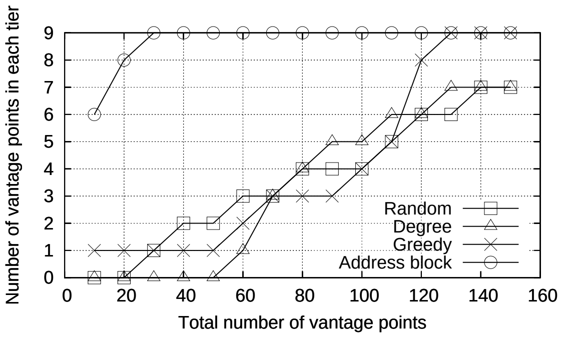


Figure 3.2: Monitors in Tier 1

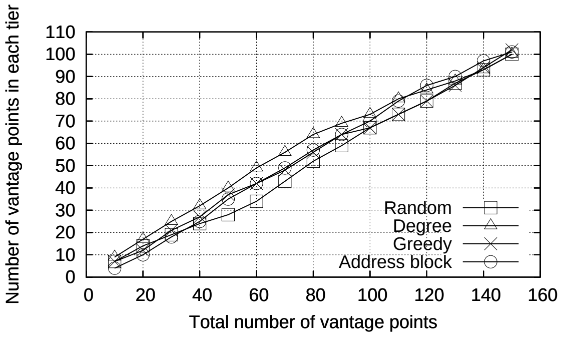


Figure 3.3: Monitors in Tier 2

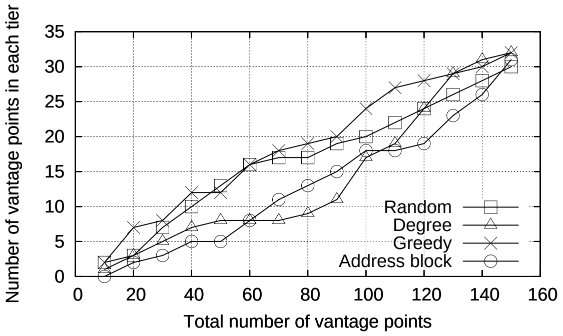


Figure 3.4: Monitors in Tier 3

3.4 Monitor Selection Analysis

In the previous section, we have observed some differences and similarities among the three realistic deployment settings. To delve deeper, we apply four simple schemes to identify the incremental benefit and even possible negative effects of adding monitors for a wider set of applications.

3.4.1 Monitor selection schemes

Our candidate set of monitors consists of all BGP feeds we have access to. We study the following four ways of adding monitors.

Random based: monitor nodes are selected randomly.

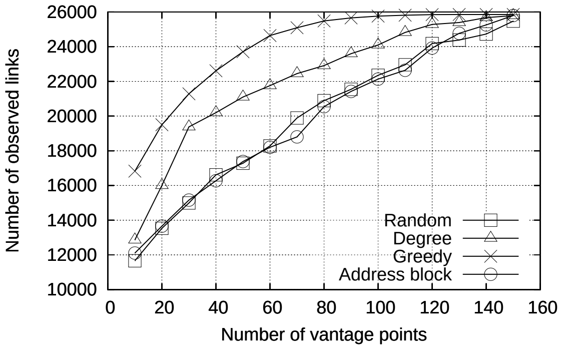


Figure 3.5: Number of observed links

Degree based: monitors with the largest node degree are selected first based on the entire data set. Node degree means the number of neighbors each AS has.

Greedy link based: at any time, the next monitor is selected with the largest number of unobserved links, given the set of already selected monitors.

Address block based: without relying on all the data, monitors in the ASes that originate the largest number of IP addresses are selected with random tie breaking.

3.4.2 Discovery of static network properties

To fully understand how each scheme works, we study the topological distribution of the monitors selected based on the tier classification, with the first three tiers shown in Figure 3.2 3.3, and 3.4. We observe that as expected the address-block-based scheme always selects the Tier-1 nodes first as they usually announce largest number of addresses. For Tier 2 and Tier 3, there is little difference among the schemes.

We first show that the observed link count increases with vantage point in Figure 3.5. Confirming previous studies [95], the increase of links from 80 vantage points can be twice as the links observed from one. The greedy-based scheme performs best as expected, followed by the the degree-based one. Interestingly, the address block based scheme is no better than random selection. This is likely due to the fact that most ASes in our candidate set contribute a similar number of links.

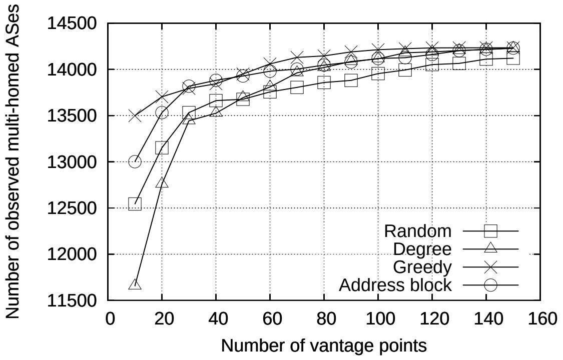


Figure 3.6: Number of observed multi-homing stub ASes

Next, we study the prevalence of multi-homing at edge networks for network redundancy as shown in Figure 3.6. The greedy-based selection again performs best as additional edge links for multi-homed stub ASes are more likely discovered. The difference between random and greedy can be up to several hundred, indicating that we may not have a complete set of multi-homed customer ASes.

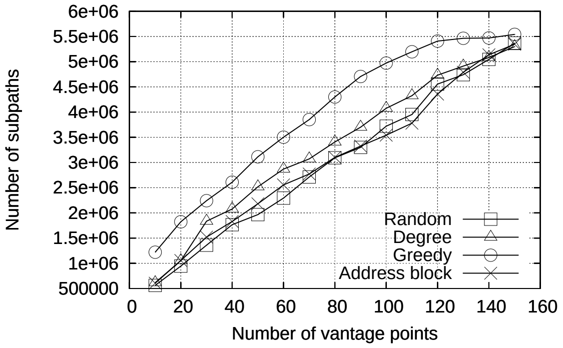


Figure 3.7: Observed AS path count (including subpaths)

As we have shown, accurate AS path prediction is still quite challenging. One way to lower the difficulty is to collect as many empirically observed AS paths as possible, as depicted in Figure 3.7. Greedy performs the best, followed by the degree-based scheme. Note that the absolute difference in observed paths for the same number of vantage points among various schemes can be as large as one million.

3.4.3 Discovery of dynamic network properties

We study two applications relying on monitoring of dynamic routing events.

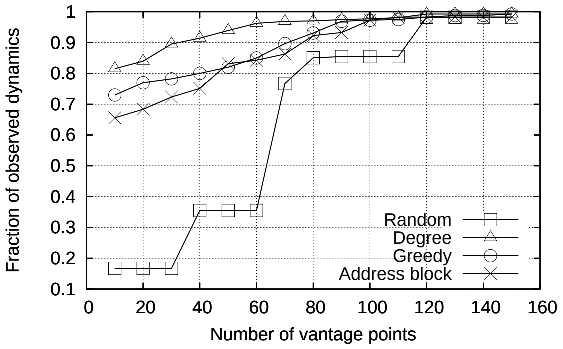


Figure 3.8: Fraction of observed routing events

Routing instability monitoring: A single network event such as link failure can trigger routing updates from many networks. We study how to monitor as many routing events occurring on the Internet as possible. Figure 3.8 shows the fraction of BGP routing events observed by the set of vantage points selected. Notice there is a huge difference between random selection and the other three schemes, indicating that vantage points associated with core networks (i.e.with high degree and many links, and originating many addresses) are more likely to observe network instabilities.

IP prefix hijacking detection: Intuitively, more monitors enable more diverse paths to be observed. Therefore, the IP prefix hijacking detection system has a higher chance of detecting all hijacks. However, based on our simulations, we observe there still exist attacker-victim pairs that can evade detection even using all the monitoring

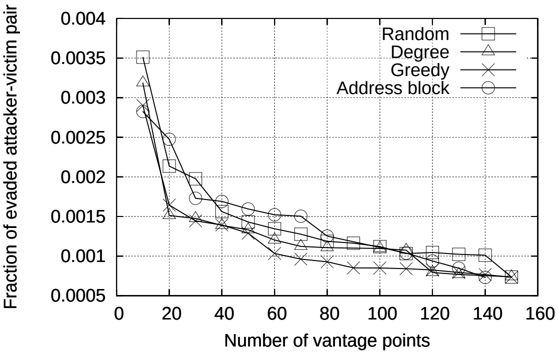


Figure 3.9: Evasion of prefix hijacking detection: number of attacker-victim pairs

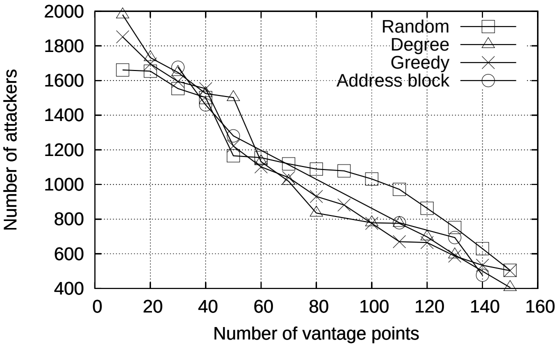


Figure 3.10: Evasion of prefix hijacking detection: number of attackers per victim

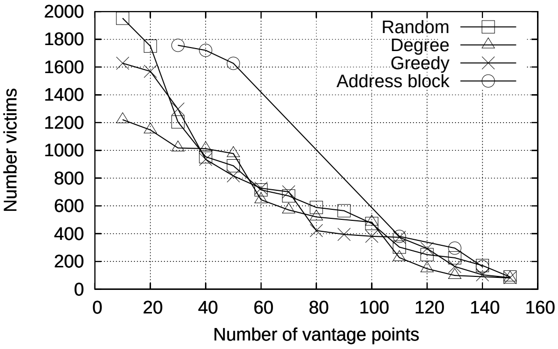


Figure 3.11: Evasion of prefix hijacking detection: number of victims per attacker

feeds we have access to. Studying to what extent attackers can evade detection is important for knowing the limitation of current detection systems due to visibility constraints.

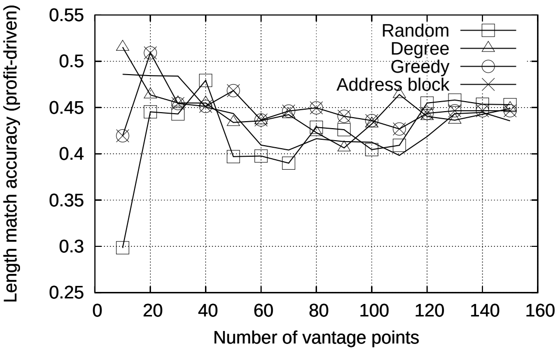


Figure 3.12: Profit-driven path prediction accuracy (length match).

The main metric we study is the number of attacker-victim pairs that can evade detection. As shown in Figure 3.9, with 10 nodes deployed in the random scheme, 0.35% of all possible attacker-victim pairs can evade the detection, which is the worst case we observe from our simulation. We also show changes in the average number of evading attackers for each victim in Figure 3.10, and in the average number of victims an attacker can attack without being detected in Figure 3.11. Overall, address block scheme performs similar to the random scheme, while greedy performs the best in most cases.

3.4.4 Inference of network properties

In the following we analyze the effect of vantage point selection on inference of AS relationships and AS-level paths. We study two algorithms for path inference.

3.4.4.1 AS relationship inference and path prediction

We first study commonly used path inference algorithms relying on AS relationships as indicated in Table 3.1. In particular, we apply Gao's degree-based relationship inference scheme [56] and then predict paths enforcing the AS relationships.

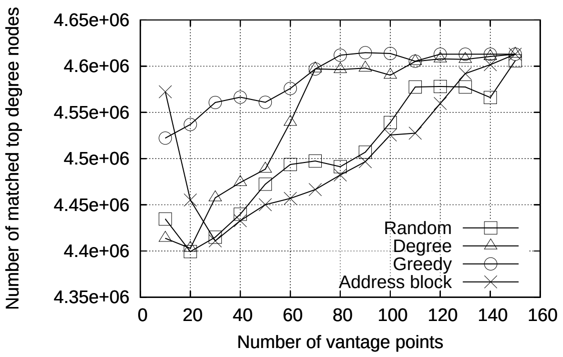


Figure 3.13: Number of matched top degree AS in all observed AS paths

Figure 3.12 shows that, surprisingly, as the number of monitors increases, the accuracy may decrease compared with observed AS paths.

We conjecture this may be caused by the nature of the degree-based relationship inference algorithm. The algorithm determines the AS relationships based on the relative degree values of AS nodes within an AS path. The topology obtained from the vantage points tends to be quite complete already in terms of relative degree information. As more vantage points are added, more noise may be introduced causing inaccuracies in inferred AS relationships.

To further understand this, we analyze the changes in the top degree node per path to explain why the increase in number of vantage points does not always result in increased accuracy. Based on the degree of each node observed in the topology using all data available, we identify the top degree AS for each observed AS path. From each set of vantage points we also locate the top-degree node. We then examine for each monitor data set, the fraction of matched top ASes for all AS paths compared with the case for the complete topology, as shown in Figure 3.13. The fluctuation in the graph indicates that additional BGP data does not consistently improve the estimation of the top-degree nodes in each path.

We emphasize that we have made an important observation: BGP data from more vantage points may not necessarily increase the accuracy of inferred network properties. The inference algorithm [56] is based on degree, which may vary in different selection of monitors: the further away an AS is to the monitor, the more incomplete the observed degree is. We point out that developing inference algorithms that are less sensitive to available data feeds but also more fully utilize the data is important in this area. We also observe that profit-driven path prediction as shown in Figure 3.12 actually performs worse than length-driven prediction. This could be due to the fact that profit-driven path selection is more sensitive to the impact caused by inaccurate AS relationship inference.

Besides accuracy, we also perform other sanity checks for inferred relationships. Two metrics are used: first, some observed paths are considered as invalid based on the inferred relationships. The fraction of such invalid paths can be used as an indication of inaccurate AS relationship inference. We found that the number of observed invalid paths slightly decreases as the number of vantage point increases. Second, for some node pairs no valid paths are predicted. Such disconnected node pairs can be used as another metric of relationship inference inaccuracy. The number of invalid paths generally decreases with more vantage points as expected; similarly, the number of AS pairs with valid paths increases with vantage point. Greedy is again observed to be the best for identifying valid paths.

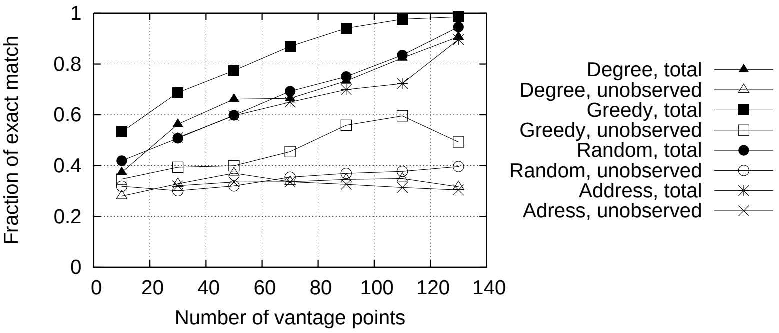


Figure 3.14: Sampled path prediction accuracy: exact matching (new algorithm)

3.4.4.2 AS-relationship-independent path prediction

In the following, we study the behavior of a recent proposed path prediction algorithm [89] that does not rely on AS relationships for prediction. For each deployment scenario, we use all observed AS paths to construct an initial topology model, and then use observed AS paths of 50 random prefixes to iteratively train the topology model using the refinement algorithm specified. The trained model is used to predict the paths from any AS to the same 50 prefixes.

To evaluate the accuracy of the predicted paths, we consider three sets of paths. The first set, *total*, is the AS paths to the 50 prefixes observed from the total default-free 165 vantage point ASes. The second set *observed* is the AS paths to the 50 prefixes observed from all the monitors a particular deployment scenario. The third set *unobserved* is the complementary set of *observed* in *total*. The algorithm always produces a perfect match on the *observed* set. Therefore, we use the other two sets for evaluation. Note that the path prediction in Section 3.4.4.1 is evaluated on *observed* instead.

Figure 3.14 shows the fraction of paths in *total* and *unobserved* that match the predicted paths. Overall, all schemes accurately predict 28% $\sim$ 60% of the unobserved paths in all scenarios. This number is lower than those in [90] because we do not include suffix subpaths in the evaluation sets, and hence do not give partial credits to the paths that partially match the prediction. The match percentage on *unobserved* generally does not increase with more monitors. The above observations show the difficulty of path prediction: predicting an unobserved path does not benefit much from observing its subpaths or its reverse path. The figure also shows that the accuracy on the *total* set improves with more monitors, which is a result of more paths being observed. Greedy performs best on the *total* set because this scheme observes most paths.

3.5 Summary

Understanding the limitation of route monitor deployment is critical for any system relying on BGP data from multiple vantage points. This understanding also enables us to better interpret the findings of previous research in terms of their generality and representativeness. Note that it is impossible to obtain routing data in real time from every network due to the scalability issue and privacy concern. Moreover, a single BGP feed from one AS also presents a restricted view given there are many routers in an AS, each with a potentially different view of routing dynamics. However, for the purpose of detecting routing anomalies, traffic engineering, topology discovery and other applications, it is useful to have additional feeds. But adding an additional feed usually requires interacting with a particular ISP to set up the monitoring session. Thus, an urgent question is to understand how to select monitor locations to maximize the overall effectiveness of the route monitoring system.

In this chapter we illustrate the importance of route monitor selection on various applications relying on BGP data. We study BGP data's impact on three categories of applications, namely, (1) discovery of relatively stable Internet properties such as the AS topology and prefix to origin AS mappings, (2) discovery of dynamic routing behavior such as IP prefix hijack attacks and routing instability, and (3) inference of important network properties such as AS relationships and AS-level paths. For each cateogy, we study various monitor deployment strategies by choosing ASes with diverse topological properties.

We summarize our key results in the following. For the first class, more vantage points generally improve completeness and accuracy of the topological properties studied. we find that larger set of monitors can observe much more links but only slightly more non-private ASes. The additional ASes identified are mostly at the edge. Using Gao's degree-based relationship inference algorithm, we compare the accuracy of inferred paths comparing with paths in BGP data in terms of path length. We found the improvement is small for path prediction with increasing vantage points. These results imply that Gao's algorithm is reasonably stable with changes in the BGP data. For routing instability detection, we found a huge difference between different schemes, indicating that vantage points associated with core networks are more likely to observe network instabilities. For attack evasion, we show that it is important to take into consideration possibility of evasion due to visibility constraints for detecting routing attacks.

Overall, our work demonstrates the limitation of purely relying on the ISP data.

It motivates future work in the area of building monitoring and diagnosis systems without ISP proprietary purely from end hosts. Revisiting the BGP based monitoring described in Chapter II, all those studies relying on BGP routing data usually assume that data from the route monitoring systems is reasonably representative of the global Internet. Our work studied the limitations of route monitoring systems and the visibility constraint of different deployment scenarios. We are the first to point out the monitor location's limitation on the attack detection. It suggests that any detection system should be aware of the detection inaccuracy induced by vantage point constraints.

This work also suggests an inherent limitation of approaches relying on routing data alone. Given that most ISPs are reluctant about revealing details of their networks, they normally keep their routing feeds publicly inaccessible. The existing public routing data repositories, RouteViews and RIPE, receive data from only around 154 ISPs, in most cases with one feed from each AS. The results in this chapter show that sometimes it is insufficient to detect routing events, not to mention locating the failure to a particular ISP. Given this fundamental limitation, in Chapter IV we investigate the techniques to detect and locate performance disruptions using an end host based approach.

CHAPTER IV

Diagnosing Routing Disruptions from End Systems

4.1 Introduction

The end-to-end performance of distributed applications and network services are susceptible to routing disruptions in ISP networks. Recent work has found routing disruptions often lead to periods of significant packet drops, high latencies, or even temporary reachability loss [51, 103, 123, 138]. The ability to pinpoint the network responsible for observed routing disruptions is critical for network operators to quickly identify the cause of the problems and mitigate potential impact on customers. In response, operators may tune their network configurations or notify other ISPs based on whether routing disruptions originate from their internal networks, their border routers, or remote networks. They may also find alternate routes or inform affected customers for destinations which will experience degraded performance.

From end users' perspective, the ability to diagnose routing disruptions also provide insight into the reliability of ISP networks and ways to improve the network infrastructure as a whole. Knowing which ISPs should be held accountable for which routing disruptions helps customers assess the compliance of their service-level agreements (SLAs) and provides strong incentives for ISPs to enhance their service quality.

Past work on diagnosing routing events has relied on routing feeds from each ISP. These techniques have proven to be effective in pinpointing routing events across multiple ISPs [54] or specific to a particular ISP [125]. However, given that most ISPs are reluctant about revealing details of their networks, they normally keep their routing feeds publicly inaccessible. Today, the largest public routing data repositories, RouteViews and RIPE, receive data from only around 154 ISPs [10, 14], in most cases with one feed from each AS. These have shown to be insufficient to localize routing events to a particular ISP in most cases [119]. As a result, customers are in the dark about whether their service providers meet their service agreements. Similarly, ISPs have limited ways to find out whether the problems experienced by their customers are caused by their neighbors or some remote networks. They usually have to rely on phone calls or emails to perform troubleshooting [8].

Motivated by the above observations, we aim to develop new techniques for diagnosing routing events from end systems. End systems are effectively hosts end-users have access to and are typically located at the edge of the Internet. Our approach differs markedly from recent work on pinpointing routing events in that it purely relies on probing launched from end-hosts and does not require any ISP proprietary information. In fact, using active probing on the data plane, our system can more accurately measure the performance of the actual forwarding paths used rather than merely knowing the expected routes to be used based on routing advertisements. Furthermore, our techniques can be easily applied to many different ISPs instead of being restricted to any particular one. This is especially useful for diagnosing inter-domain routing events which often requires cooperation among multiple ISPs. Our inference results can be made easily accessible to both customers and ISPs who need better visibility into other networks. This is also helpful for independent SLA monitoring and routing disruptions management stemmed from other networks. In addition, end system probing can be used for both diagnosing and measuring the performance impact of routing events. It offers us a unique perspective to understand the impact of routing events on end-to-end network performance.

In this chapter, we consider the problem of diagnosing routing events for any given ISP based on end system probing. Realizing that identifying the root cause of routing events is intrinsically difficult as illustrated by Teixeira and Rexford [119], we focus on finding explanations for routing events that the ISP should be held accountable for and can directly address, e.g.internal routing changes and peering session failures. In essence, we try to tackle the similar problem specified by Wu [125] without using ISP's proprietary routing feeds. Given that end systems do not have any direct visibility into the routing state of an ISP, our system overcomes two key challenges: i) discovery of routing events that affect an ISP from end systems; and ii) inference the cause of routing events based on observations from end systems. We present the details of our approach and its limitations in terms of coverage, probing granularity, and missed routing attributes in Session4.3.

We have designed and implemented a system that diagnoses routing events based on end system probing. Our system relies on collaborative probing from end systems to identify and classify routing events that affect an ISP. It models the routing event correlation problem as a bipartite graph and searches for plausible explanation of these events using a greedy algorithm. Our algorithm is based on the intuition that routing events occurring close together are likely caused by only a few causes, which do not create many inconsistencies. We also use probing results to study the impact of routing events on end-to-end path latency.

We instantiate our system on PlanetLab and use it to diagnose routing events for five big ISPs over a period of more than three and half months. Although each end-host only has limited visibility into the routing state of these ISPs, our system is able to discover a large number of significant routing events, e.g.hot-potato changes and peering session resets, during that period. We validate the accuracy of our inference results in two ways. Comparing with existing ISP-centric method, we are able to distinguish internal and external events with up to 91.2% accuracy. We are able to identify 4 out of 6 disruptions reported from NANOG mailing lists [8].

We summarize our main contributions. Our work is the first to enable end systems to scalably and accurately diagnose causes for routing events associated with large ISPs without requiring access to any proprietary data such as real-time routing feeds from many routers inside an ISP. Unlike existing approaches to diagnose routing events associated ISPs, our approach of using end system based probing creates a more accurate view of the performance experienced by the data-plane forwarding path. Our work is a first step to enable diagnosis of routing disruptions on the global Internet accounting for end-to-end performance degradations.

The rest of this chapter is organized as follows. We describe the overview of system architecture in Session4.2, followed by description of the collaborative probing in Session4.3. Session 4.4 illustrates the procedure to identify individual routing changes. Then in Session 4.5 we discuss the algorithm for root cause inference. The deployment results are shown in Session 4.6 with validation shown in Session 4.7.

4.2 System Architecture

We present an overview of our system in this section. To diagnose routing events for any given ISP (which we call **a target ISP**), our system must learn the continuous routing state of the ISP. Based on the change in routing state, it identifies and classifies individual routing events. Because a single routing disruption often introduces many routing events, our system applies an inference algorithm to find explanation for cluster of events occurring closely in time. It then uses the latency measurements in the probes to quantify the impact of these routing events. As shown in Figure, our system is comprised of four components:

Collaborative probing: This component learns the routing state of a given ISP via continuous probing from multiple end systems. Given the large number of destinations on the Internet, the key challenge is to select an appropriate subset to ensure coverage and scalability.

Event identification and classification: This component identifies routing events from a large number of end system probes. These events are then classified into several types based on the set of possible causes, e.g.internal change, peering failure, or

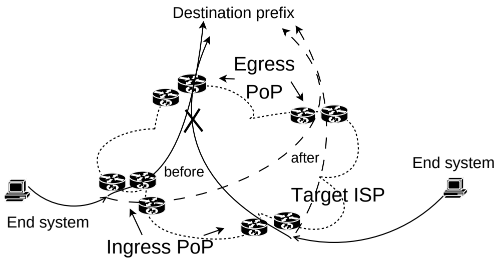


Figure 4.1: Collaborative probing to discover routing events.

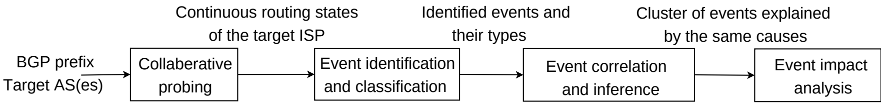


Figure 4.2: System Architecture

external change.

Event correlation and inference: This component searches for plausible explanation for routing events. Although each routing event may be triggered by many possible causes, we seek to identify a small set of causes that can explain all the events occurring close in time. We model the inference problem as a bipartite graph and solve it with a greedy algorithm.

Event impact analysis: This component extracts latency information from end system probes. It enables us to study the impact of routing events on path latency according to the cause of events and the impacted ISPs. Note that this information is not readily available in routing feeds used in previous work on routing diagnosis.

4.3 Collaborative Probing

For a target ISP, we need to know its routing state to identify and diagnose its routing events. Unlike previous work that uses many routing feeds from a single ISP [125], our system relies on end systems that do not have any direct visibility into ISP's routing state. Note that it is important to obtain a comprehensive view of the routing state across major Points of Presence (PoPs) of the target ISP in order to diagnose routing events associated with the ISP. Utilizing public routing repositories, aside from the fact they are not real-time, is insufficient due to only one or at most two feeds from each ISP. The key question in our design is how we can learn the routing state of an ISP from end system probing alone.

4.3.1 Learning routing state via probing

A router's routing table contains the traffic forwarding information, e.g.the next hop, based on the destination prefix. Although an end system may not have direct access to the routing tables, it could learn this next hop information using *traceroute* if the forward path from the host to the destination happens to traverse the router. As is illustrated in Figure 5.2, traceroute probing from two end systems to one particular destination experience egress PoP shifts due to the target ISP's internal disruption. Ideally, we can learn the next hop from any router to any destination by probing from an appropriate source. This may not always be possible because we may not have access to such a source or the router may not respond to our probes.

We focus on diagnosing inter-domain routing events that affect a target ISP. We aim to find explanations for events that the ISP should be held accountable for and can directly address, e.g.internal routing changes and peering session failures. For internal or intra-domain routing events it is obvious which ISP should take responsibility for them. Therefore, we do not focus on constructing detailed intra-domain routing tables. Instead, we keep track of the inter-domain routing tables (BGP tables) of each major PoP within the ISP.

There are three challenges associated with constructing BGP tables. First, given a limited set of end systems, the system attempts to obtain as many routes between PoP-prefix pairs (PoP to destination prefix) as possible. Second, end systems have limited resources (CPU and network), and our system must have low probing overhead. Third, probing needs to be launched frequently to accurately track the dynamic routing state.

To address the first two challenges, we devise a scheme to select an appropriate set of destinations for each end system to probe. We start with a set of prefixes extracted from BGP tables. Each end system acquires its own routing view by conducting traceroute to one IP in each of these prefixes. Using the existing method developed in Rocketfuel [110], we can infer whether each traceroute probe goes through the target ISP and the PoPs traversed. Combining the routing views from all the end systems, we obtain a complete set of PoP-prefix pairs visible to our system. We then try to select a minimum set of traceroute probes that can cover all the visible PoP-prefix pairs with a greedy algorithm. At each step, we select a traceroute probe that traverses the maximum number of uncovered PoP-prefix pairs and remove these newly-covered pairs from the set of uncovered pairs. This process continues until there is no uncovered PoP-prefix pair left. The scheduling process has been shown to be effective in balancing between coverage and overhead [83]. Note that because ISP network topology and routing evolve over time, each end system periodically refreshes its routing view. Currently, this is done once a day to achieve a balance between limiting probing overhead and capturing long-term changes.

To address the third challenge, we developed a customized version of traceroute which enhances the probing rate by measuring multiple destinations and multiple hops in parallel up to a pre-configured maximum rate. To prevent our measurement results from being affected by load-balancing routers, all probe packets have the same port numbers and type of service value. With our improvement, all the end systems can finish probing their assigned set of destinations in roughly *twenty minutes*. This also means our system can obtain a new routing state of the target ISP every twenty minutes, the details of which will be shown in §4.6.

4.3.2 Discussion

Although learning ISP's routing state via collaborative probing does not require any ISP proprietary information, it has three major limitations compared with directly accessing BGP routing feeds: (i) given a limited number of end systems, we cannot learn the route for every PoP-prefix pair; (ii) given limited CPU and network resources at end systems, we cannot probe every PoP-prefix pair as frequently as desired. This implies we may miss some routing events that occur between two consecutive probes; and iii) we can only observe routing changes but not the other

BGP attributes changes.

The first problem is a common hurdle for systems focused on finding root causes of routing changes as described by Teixeira and Rexford [119]. They presented an ideal architecture for cooperative diagnosis which requires coverage in every AS. Similar to the work by Wu *et al.*, our work addresses a simpler problem of diagnosing routing changes associated with a large ISP but purely from end system's perspectives. Our ability to solving this problem relies on the coverage we get.

A straightforward solution to improving coverage is to use more end systems. We use all the available PlanetLab sites (roughly 200) to probe five target ISPs. We will explain the detailed coverage results in §4.6. Note that a single major routing disruption near the target ISP, e.g.a hot-potato change or a peering session failure, often introduces a large number of routing events and affects many different PoPs and prefixes. In §4.7, we will show that our system is able to correctly identify many such major disruptions despite covering only a subset of the affected PoP-prefix pairs. As future work, we plan to study to how the coverage will improve our inference accuracy.

We consider the second problem to be less important. Our system focuses on diagnosing routing changes that are long-lived enough to warrant ISP's corrective action rather than those that are transient and repaired by itself quickly. The latter may overwhelm the ISPs. Given that inter-domain routing changes can require up to several minutes to converge [74], our system is sufficient to detect the significant changes.

The third problem is more fundamental to systems that rely on end system probing, given that such BGP information is inherently proprietary. This implies we might identify or locate a routing change but might not know *why* it occurs. We give an example of this in §4.5 where we cannot distinguish a route change triggered by different attribute changes. The focus of our work is on determining whether an ISP should be held accountable for a routing problem and provide useful hints for the ISP to diagnose it. After this is done, we believe the ISP itself has the most knowledge to find the root-cause.

4.4 Event Identification and Classification

In this section, we first describe how we identify individual routing events from the time sequence of routing states captured for a target ISP. We then present our event classification method based on likely causes.

As explained in the previous section, we focus on the inter-domain routing state of the target ISP. Given a PoP-prefix pair, we identify the next hop and the AS path from the PoP to the destination prefix. The next hop can be either a PoP in the target ISP or another ISP. This implies we need to extract the ISP and PoP information from end systems' traceroute probes.

A traceroute probe only contains the router's interface addresses along the forwarding path from the source to the destination. We map an IP address to a PoP in the target ISP using the existing tool based on DNS names (*undns*) [112]. For instance, 12.122.12.109 reverse-resolves to *tbr2-p012601.phlpa.ip.att.net*, indicating it is in AT&T network, located in Philadelphia (phlpa). *undns* contains encoded rules about ISPs' naming convention. For IP addresses not in the target ISP, we map them to ASes based on their origin ASes in the BGP tables [87]. One IP address may map to multiple origin ASes (MOAS) and we keep a set of origin ASes for such IP addresses. After performing the IP-to-PoP and IP-to-AS mappings for each traceroute probe, we know the traversed PoPs in the target ISP and the AS path to destination prefix. The IP-to-AS and IP-to-PoP mapping may have inaccuracies. Given that errors in IP-to-AS and IP-to-PoP mappings are sometimes inevitable, we present a method to deal with such errors in event correlation and inference (§4.5). The errors in these mappings are less likely to affect the root cause inference accuracy since we use a greedy algorithm for correlation which will more likely uses the correct mappings.

Note that not all the traceroute probes are used for routing event identification and classification. They may be discarded for several reasons:

Traceroute probes may not traverse the target ISP when the source hosts do not have up-to-date routing views or the probes are conducted during temporary routing changes. Such probes are discarded because they do not contribute any routing information about the target ISP.

Traceroute paths may contain '*' hops when routers do not respond to probes due to ICMP filtering or rate-limiting. A '*' hop is treated as a wildcard and can map to any ISP or PoP. To simplify path matching for event identification, we discard traceroute containing two or more contiguous '*' hops.

Traceroute paths may contain transient loops likely capturing routing convergence. Such traceroute paths are not stable and somewhat arbitrary because they depend on the subtle timing when routers explore alternate paths. Since our goal is to infer the likely causes of routing events, we are interested in the stable paths before and after a routing event rather than the details of the transition. We discard traceroute paths that contain IP-level, PoP-level, or AS-level transient loops.

Some traceroute paths may contain loops that persist for more than 20 minutes. Since most routing convergence periods are within several minutes [74], these loops are likely caused by routing anomalies [127] rather than the unstable router state during convergence. We still make use of such traceroute paths after truncating their loops, since these represent stable paths.

Traceroute paths could be incomplete due to ICMP filtering or rate limiting, which appears to be stars in certain hops. We consider any paths containing two consecutive stars as incomplete, as missing one hop usually can be inferred from comparison with previous paths. To save some data, we only discard the paths with two consecutive starts in the IP level path inside and after the target AS, as we focus on identifying the changes from the target AS to the destination. We also discard the paths containing consecutive starts in the PoP level path and the AS level paths. The AS level path is actually the sub paths after the target AS.

One special type of incomplete traceroute is due to ICMP filtering, which makes the traceroute paths stop before entering the destination AS. This observation is usually persistent as the ISPs' policies are not often changed. Therefore, these paths are still used for later analysis as they are considered stable.Although they are only useful for detecting the changes before the entering the ISP performing filtering.

Our probing methodology is designed to avoid some other inaccuracies. To avoid traceroute path change due to load balancing, we use our own customized tool similar to Paris-traceroute [19].

We now describe how we identify inter-domain routing events that affect the target ISP from the continuous snapshots of routing states obtained from traceroute probes. An *inter-domain routing event* is defined as a path change from a PoP to a destination prefix, in which case either the next hop or the AS path has changed. Since our system acquires a new routing state of the target ISP periodically, we can identify an event by observing a path change between the same source and destination in two consecutive measurements.

Given that there could be '*' hops and multiple-origin-ASes (MOAS) hops, we choose to be conservative in comparing two paths by trying to search for their best possible match. For instance, $path(A,\ *,\ C)$ is considered to match $path(A,\ B,\ C)$ because '*' can match any ISP or PoP. Similarly, a MOAS hop can match any AS in its origin AS set.

When observing path changes between two consecutive measurements, we classify them into three types according to their likely causes. The categories are based on our goal of pinpointing the ISP that causes routing changes. We focus on the only commonly occurred types of routing disruptions.

Type 1: Different ingress PoP changes can be caused by routing events in the up-stream ISPs, the target ISP, or down-stream ISPs. Realizing it is difficult to enumerate all the possible causes, we do not currently use them for event correlation and inference.

Type 2: Same ingress PoP but different egress PoP changes can be caused by internal disruptions in the target ISP (e.g.hot-potato changes), failures on its border(e.g.peering session reset), or external changes propagated to the target ISP (e.g.prefix withdraw).

Type 3: Same ingress PoP and same egress PoP changes are easier to deal with compared with the previous two types. They may involve internal PoP path changes, external AS path changes, or both. We will explain how to use such information for event correlation and inference in the next section.

4.5 Event Correlation and Inference

It is well known that a single major routing disruption often leads to a burst of routing events and affects many PoPs and prefixes simultaneously. Our goal is to diagnose which inter-domain routing events are triggered by those major disruptions that the target ISP should be held accountable for and can take action on.

In many cases, it is extremely difficult to infer the cause of an individual routing event because an event may be explained by many different causes. An obvious solution is to improve inference accuracy by correlating multiple "relevant" events together. However, the key question is how we can discover and make use of the relevancy between events.

1. Ignore if the next hop is unreachable
2. Highest local preference
3. Shortest AS path
4. Lowest origin type
5. Lowest Multiple-Exit-Discriminator (MED) value among routes from the same AS
6. eBGP learned route over iBGP learned route
7. Lowest IGP cost (hot-potato)
8. Lowest router ID

Table 4.1: BGP decision process

4.5.1 Inference model

Before describing our inference model used for event correlation, we make an assumption that each routing event can be explained by only one cause. This is a standard assumption made in many existing works on root cause analysis [54, 119] and fault diagnosis [72]. Note that this assumption does not prevent us from inferring multiple simultaneous causes as long as the events triggered by different causes are independent.

We start by defining a few terminologies to facilitate our discussion. Since each event is identified by observing the change between two consecutive path probes, we call the earlier path probe an **old path** and the latter one a **new path**. We call the egress PoP on the old/new path the old/new egress respectively. In the previous section, we classify individual routing events into three types. Currently, we do not use the events of the first type for correlation because it is infeasible to enumerate all the possible causes for them. We enumerate all the possible causes for the latter two types of events based on how BGP selects a single best route for each prefix. When multiple routes are available, BGP follows the decision process in Table 4.1 to select the best one.

Same ingress PoP but different egress PoP changes can be triggered by a prefix withdraw, a prefix announce, or a change in any of the eight steps in Table 4.1. We ignore $Step_8$ since router ID rarely changes. $Step_6$ is irrelevant because both the old and the new egresses use external paths. The following causes comprehensively cover all the remaining possibilities:

- A change in $Step_1$ is explained by either an *Old-Peering-Down* or a *New-Peering-Up*. The former implies the peering between the old egress and its next hop AS is down. The latter means the peering between the new egress and its next hop AS is up.

- A change in $Step_2$ can be explained by either an *Old-Lpref-Decrease* or a *New-Lpref-Increase*. The former implies the local preference ($Lpref$) at the old egress decreases. The latter implies the $Lpref$ at the new egress increases.

- A prefix withdraw, an announce, or a change in $Step_{3-5}$ can be explained by either an *Old-External-Worsen* or a *New-External-Improve*. The former means the old route to the prefix worsens due to an external factor (e.g.a prefix withdraw, a longer AS path, a higher origin type, or a higher MED value). The latter implies the new route to the prefix improves due to an external factor (e.g.a prefix announce, a shorter AS path, a lower origin type, or a lower MED value).

- A change in $Step_7$ can be explained by an *Old-Internal-Increase* or a *New-*

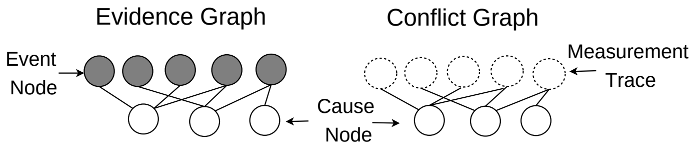


Figure 4.3: The bipartite graph of root cause inference

Internal-Decrease. The former implies the cost of the old internal path increases due to a more costly PoP-level link. The latter implies the cost of the new internal path decreases due to a less costly PoP-level link.

Same ingress PoP and same egress PoP changes

- When there is internal PoP path change, it can be explained by an *Old-Internal-Increase* or a *New-Internal-Decrease*.
- When there is next hop AS change, it can be explained by an *Old-Peering-Down*, a *New-Peering-Up*, an *Old-External-Worsen*, or a *New-External-Improve*.
- When there is AS path change but no next hop AS change, it can be explained by an *External-AS-Change*. It implies the change is not directly related to the target ISP.

Using the above rules, we can map each event to a set of possible causes. By aggregating events that occur closely in time (identified between the same pair of consecutive routing states), we construct *a bipartite graph*, called *evidence graph*, as shown in Figure 4.3. There are two types of nodes in an evidence graph: cause nodes at the bottom and event nodes at the top. An edge between a cause node and an event node indicates the event can be explained by the cause. An evidence graph encapsulates the relationship between all the possible causes and their supporting evidences (events).

Conflicts may exist between causes and measurement traces due to noise and errors. For instance, an *Old-Peering-Down* will conflict with a new trace which traverses the peering that is inferred to be down. Conflicts stem from two major sources: i) the subtle timing difference when traceroute probes from different end systems traverse the same peering or measure the same prefix; and ii) errors in the IP-to-AS or IP-to-PoP mappings.

A measurement trace will never conflict with an *Old-Internal-Increase* or a *New-Internal-Decrease* because a cost change on a PoP-level link may not prevent a path from using the link. However, a measurement trace may conflict with each of the remaining six causes:

- *Old-Peering-Down*: a new path still uses a peering that is inferred to be down.
- *New-Peering-Up*: an old path already used a peering that is inferred to be up.
- *Old-Lpref-Decrease*: a new path still uses an egress that is inferred to have a lower $Lpref$ even when there are other egresses with a higher $Lpref$.
- *New-Lpref-Increase*: an old path already used an egress that is inferred to have a higher $Lpref$ (therefore used to have a lower $Lpref$) even when there were other egresses with a higher $Lpref$.
- *Old-External-Worsen*: a new path still uses an old route to a prefix even when it is worse than a new route to the same prefix, or an old path already used a

new route to a prefix even when the old route to the same prefix was better.

- *New-External-Improve*: a new path still uses an old route to a prefix even when a new route to the same prefix is better, or an old path already used a new route to a prefix even when it was worse than an old route to the same prefix.

We encapsulate the relationship among all the possible causes and their conflicting measurement traces using a *conflict graph*, as shown in Figure 4.3. Similar to an evidence graph, it has two types of nodes: cause nodes at the bottom and measurement nodes at the top. An edge between a cause node and a measurement node indicates a conflict between the cause and the measurement traces. For each evidence graph, we construct a conflict graph accordingly by inspecting all the measurement traces in the same pair of consecutive routing states. When a measurement trace conflicts with some causes in the evidence graph, we insert a measurement node and the corresponding edges into the conflict graph.

4.5.2 Inference algorithm

We now present our inference algorithm that uses the evidence graph and the conflict graph to infer likely causes. Our inference is guided by two rules: i) simplest explanation is most likely to be true. We try to find the minimum set of causes that can explain all the evidences (events); and ii) we should take into account the noise and errors in our measurement by minimizing conflicts between inferred causes and measurement traces.

We use a greedy algorithm to infer causes. In each iteration, it selects a cause from the evidence graph with the maximum value of $(E - \alpha C)$, where E is the number of supporting events and C is the number of conflicting traces (computed from the conflict graph). Intuitively, it selects a cause that explains many events but raises few conflicts. It then removes the events that have been explained by the cause from the evidence graph before entering the next iteration. This process continues until all the events have been explained.

The parameter α allows us to tune the relative weight between evidences and conflicts. A larger α makes our algorithm more aggressive in avoiding conflicts. Currently, we set $\alpha = 1$ in our experiments. However, we find our results are not very sensitive to the choice of α between 0.1 and 10. This is likely due to the fact that the number of evidence significantly outweighs the number of conflicts for most causes (see §4.7).

Given that the inputs to our algorithm (the evidence graph and the conflict graph) are limited by the coverage of our system and measurement noise and errors, it may report incorrect causes or miss true causes. To highlight the difference between the reliability of inferred causes, we introduce a notion of *inference confidence* for each cause as $E - \alpha C$, where E and C have the same meaning as in the above. Intuitively, causes with a higher inference confidence or more evidence but fewer conflicts are more reliable. We will demonstrate how inference confidence affects our inference accuracy in §4.7.

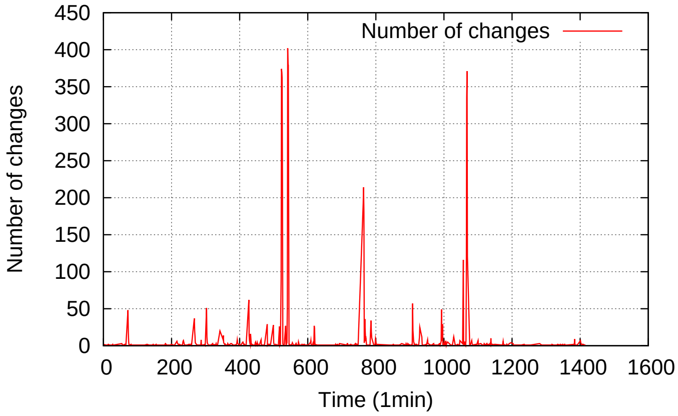


Figure 4.4: Number of changes detected (Sep. 25, 2007)

ASN (Tier)	Name	Periods	# of Src	# of PoPs	# of Probes	Probe Gap
7018 (1)	AT&T	3/23-4/9	230	111	61453	18.3 min
2914 (1)	Verio	4/10-4/22 9/13-9/22	218	46	81024	19.3 min
3320 (2)	Deutsche Telekom	4/23-5/22	149	64	27958	17.5 min
3561 (1)	Savvis	5/23-6/24	178	39	40989	17.4 min
11537 (3)	Abilene	9/23-9/30	113	11	17757	18.4 min

Table 4.2: Summary of data collection

4.6 Results of Event Identification and Classification

In this section, we present the results of event identification and classification using our framework over a period of 111 days for five backbone ISPs. We validate the identified routing events using BGP data from many vantage points at the end of the section.

The summary of data collection is shown in Table 7.1. We study three Tier-1 ASes, one Tier-2, and one Tier-3 AS. As the first step, we study one AS at a time. We plan to study multiple ASes simultaneously in the future to better diagnose routing events at a global scale. Table 7.1 shows the number of probing source hosts used and the number of PoPs covered. Note that there is some variability across the number of source hosts used as not all hosts are useful for improving the coverage in PoP-prefix pairs. This provides room for probing multiple ASes at the same time. We verified our PoP coverage completeness using data from Rocketfuel [110] and router configuration files from the Abilene network. Table 7.1 also shows the average number of probes to acquire the routing state of a target ISP. Depending on the ISP, each source host has to probe 187 to 371 destinations on average. As expected, our system can refresh the routing state roughly every eighteen minutes. The coverage will increase as the number of available probing end-systems increases.

Before going into the detailed description, we first use one example to illustrate that our system is able to detect significant network disruptions with a large number of routing events. Figure 4.4 shows the number of routing events detected using our system for Abilene over time on Sep. 25, 2007. It is obvious that the routing event occurrence is not evenly distributed. We do observe a few spikes across the day. The constant background noise is often due to routing events that only affect individual prefixes. The spike around 540min is because of an internal disruption causing the egress PoP shifted from Washington DC to New York, affecting 782 source-destination pairs. The next spike around 765min is because one neighbor AS2637 withdrew routes to 112 prefixes from the Atlanta PoP. The last spike around 1069min is due to a peering link failure, resulting in the next hop AS of the Washington DC PoP

	IP loop	PoP loop	AS loop	IP star	PoP star	AS star	No targetAS	Persistent IP loop	Persistent AS loop
Removed traces (%)	12584 0.18%	9906 0.14%	1013 0.015%	14055 0.2%	5832 0.08%	9473 0.13%	2466917 3.2%	1727 0.02%	410 0.005%

Table 4.3: Statistics of data cleaning: avg number of removed traces per day for each type of anomalous traceroute.

changed from AS1299 to AS20965. All these causes have been confirmed with the BGP and Syslog data of Abilene. It demonstrates the ability of our system to capture significant network disruptions.

4.6.1 Data cleaning process

As mentioned in §4.4, we first need to remove the noise in our dataset. Table 4.3 shows the overall statistics of average daily traceroute probe traces removed due to various reasons. It is expected that a relatively small percentage (0.75%) of traces are ignored due to contiguous '*' hops and temporary loops. We also found 0.025% of the traces contain persistent IP or AS loops usually occurring close to the destination, which confirms the observations from a previous study [127].

Note that 3.2% of the traces are discarded due to not traversing the target ISP. Such traces are not useful as we cannot distinguish between whether it is due to the target ISP losing reachability or any of the preceding ISPes changing routes. One noteworthy observation is that 35% of the traces stop before entering the destination. Most of these networks appear persistently unreachable over time, possibly due to ICMP filtering at the edges between provider and customers. We still use these traces as they can help detect routing changes in the partial path before entering the

Target AS	Total events (% all traces)	Ingress same Egress change	Ingress same, Egress same		Ingress change
			internal pop path	external AS path	
7018	277435	33325	213562 , 76.9%		30548
	0.35%	12.1%	51%	35%	11%
2914	415778	113507	261525 , 62.9%		40746
	0.31%	27.3%	48%	19%	9.8%
3320	437125	21419	384233 , 87.9%		31473
	0.66%	4.9%	8.5%	80.7%	7.2%
3561	311886	34307	233915 , 75%		43664
	0.35%	11%	45%	31%	14%
11537	81182	12177	45462 , 56%		23543
	0.31%	15%	37%	40%	29%

Table 4.4: Statistics of classification

destination network.

4.6.2 Identified events and their classification

We first classify routing events according to the ingress and egress PoP changes. Table 4.4 shows the fraction of events compared to all the traces as well as their distribution for all the time periods of each target ISP. Only very small fraction of the traces contain routing changes. Among these changes, a small fraction (7.2% - 29%) of them are found to be ingress PoP changes. This is because most of the probing sources enter the target AS from an ingress PoP near its geographic location. The majority (56% - 87.9%) of the events are in the category of both ingress and egress stay the same. This category contains either internal PoP-level path changes and/or the external AS path changes. The remaining events (4.9% - 27.3%) involve egress PoP changes. Some of these events may impose significant impact on the target ISP as a large amount of traffic to many prefixes shift internal paths simultaneously.

ASN	Dst. prefix coverage	Dst. traversing PoPs with BGP	Detected events AS , nexthop	Missed events short dur., filter, others
7018	34145 (15%)	3414 (1.5%)	64714, 11% (10.3%, 3.2%)	89% (75%, 13%, 1%)
2914	40881 (18.6%)	40039 (18.1%)	73689, 23% (19.1%, 8.6%)	77% (73%, 4%, 0%)
3561	17317 (7.8%)	2317 (1.1%)	55692, 6% (5.8%, 0.5%)	94% (80%, 9%, 5%)
11537	13789 (6%)	13789 (6%)	36853, 17% (13.9%, 4.9%)	83% (64%, 15%, 4%)

Table 4.5: Validation with BGP data for routing event identification and classification.

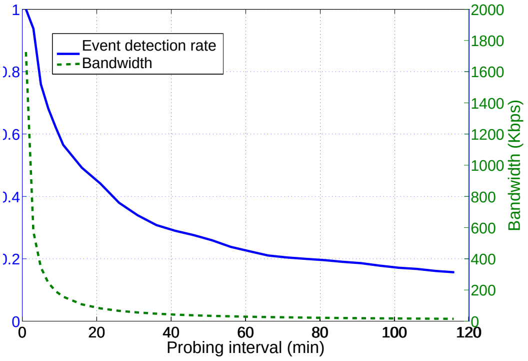


Figure 4.5: Detection ratio changes with probing interval and bandwidth

Abilene, the educational backbone network, was expected to be more stable due to its simpler topology and less frequent traffic engineering. Surprisingly, we found that it has a larger fraction of ingress changes. This is observed mainly by three source hosts, switching their ingress PoP to various destinations. Two of them are universities in Oregon, with access links to Abilene in both Seattle and Los Angeles. The other one is a university in Florida, which has access links in both Atlanta and Kansas City. We confirm this via Abilene border routers' configuration files. We believe this could be due to load-balancing or traffic engineering near the sources.

4.6.3 Validation with BGP data

Using public BGP feeds from RouteViews, RIPE and Abilene, in addition to 29 BGP feeds from a Tier-1 ISP, we validate our results in two aspects: the coverage in routing state and the coverage in detected changes. We omit AS3320 here due to lack of BGP feed from it.

To evaluate the coverage of our dataset, we map the destination IP to the longest prefix using the latest routing table for each AS. Then by comparing the set of prefixes with all the prefixes in the default-free routing table for each target AS, we compute the coverage, as shown in the second column of Table 4.5. Arguably the coverage is limited, between 6% to 18%. However, our traces cover all the known distinct PoP-level links within each target AS (compared to Rocketfuel data [110]) to detect significant routing changes associated with the AS. We didn't include the internal events in Table 4.5 as they are not able to be observed in BGP data.

We use the following methodology for validating changes detected using BGP data. For the five ASes we studied, we only have BGP feeds for four ASes. For each of them, we first identify the corresponding PoP where the BGP feed comes from. Because different PoPs in an AS usually experience different routing changes, we compare BGP-observed changes with traceroute-observed changes only when our traces traverse the PoP of the BGP feed. The third column of Table 4.5 shows the coverage of the probed destination prefixes that traverse the PoP of the BGP feed relative to the total number of prefixes for each AS.

The subset of destinations which can be used for comparison varies across ASes due to the different number of available BGP feeds. We focus on examining for any BGP-observed routing change of this subset of destinations, whether we also detect it using our traces. Moreover, we only account for BGP routing changes with either AS path changes or next hop AS changes, which can be detected via traceroute. By comparing the two sets, we calculate the fraction of changes our system can detect, as shown in the fourth column of Table 4.5. This coverage varies between 6% to 23%. Note that we can also detect many internal changes which are not observed in the BGP data (thus not included in this table).

For those changes missed by our system, there are two main reasons. First, the routing changes last too short to be detected by our two consecutive probes, accounting for the majority of the missed routing events. As explained in §4.3, we do not focus on these short-lived routing events. Second, because the traceroute may be incomplete due to packet filtering, certain changes cannot be detected as the changing path segment is invisible from our probes. Most filterings happen in the path segment after the next hop AS and close to the destination AS. Since we only use the next hop AS information for event correlation, missing these changes does not have any impact on our inference results. As is shown in Figure 4.5, the event detection rate increases with the system's maximum bandwidth consumption.

Only a small fraction (up to 5%) of the missed changes are due to other factors, e.g.inaccurate IP-to-AS mappings or mismatched forward paths compared to BGP data. In summary, our system is able to capture most routing changes to the probed destinations that are useful for event correlation and inference.

AS	Old-Int.-increase	New-Int.-decrease	Old-Peering-Down	New-Peering-Up	Old-Ext.-worsen	New-Ext.-improve	Old-Lpref-decrease	New-Lpref-increase	Ext.-AS-change
7018	5223 (4.5%)	3843 (3.3%)	5677 (4.9%)	4955 (4.3%)	18142 (16%)	20961 (18%)	302 (0.2%)	397 (0.3%)	55216 (48%)
2914	10366 (5.4%)	8135 (4.3%)	6666 (3.5%)	7024 (3.7%)	38748 (20%)	49075 (26%)	124 (0.06%)	164 (0.08%)	69190 (36%)
3320	1622 (0.5%)	954 (0.2%)	20751 (5.4%)	10204 (2.6%)	80385 (21%)	81761 (21%)	751 (0.2%)	1002 (0.2%)	185683 (48%)
3561	4410 (3.6%)	4007 (3.2%)	6017 (4.9%)	7667 (6.3%)	23232 (19%)	45495 (37%)	85 (0.07%)	105 (0.08%)	30540 (25%)
11537	750 (1.3%)	548 (0.09%)	2355 (4.1%)	909 (1.6%)	20745 (36%)	23502 (42%)	48 (0.08%)	51 (0.09%)	7434 (13%)

Table 4.6: Statistics of root cause inference.

4.7 Results of Event Correlation and Inference

In this section, we first present the results of our inference algorithm. Then we validate our system in three ways: comparing with the BGP feed based inference using BGP data from a Tier-1 ISP; comparing with both BGP data and Syslog data from the Abilene network; and comparing with disruptions reported from the NANOG email list [8].

4.7.1 Result summary

Our inference algorithm takes the set of identified events and automatically clusters them based on their causes. Table 4.6 shows both the total number and the relative percentage for each type of causes inferred for each ISP. We observe that different ISPs can have a non-negligible difference in the cause distribution. For example, for the first three ISPs, the largest fraction of events are caused by *External-AS-Change*. In contrast, Abilene (AS11537) has more events caused by *Old-External-Worsen* and

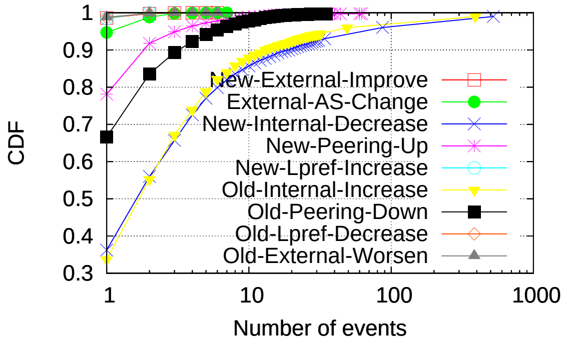


Figure 4.6: CDF of the number of events per cluster

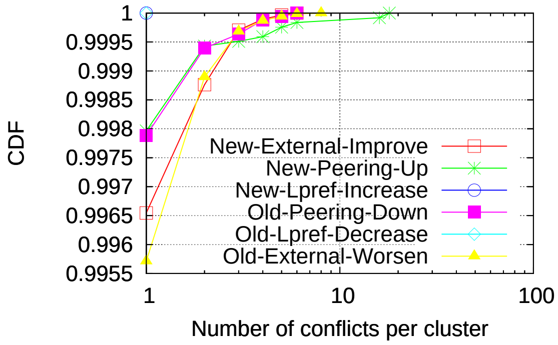


Figure 4.7: CDF of the violations per cluster

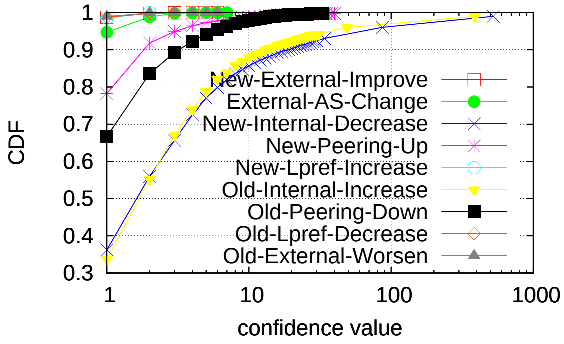


Figure 4.8: CDF of the confidence per cluster

New-External-Improve. This is mainly caused by five neighbor ASes. The most dominant one is the neighbor AS20965 peering in New York who switches routes to around 390 destinations frequently over time.

We study the effectiveness of our inference algorithm in clustering related events together in Figure 4.6. A cluster is defined to be the set of events explained by a single cause. The figure shows the CDF of the number of events per cluster over the entire period for five ASes. While most of them have less than 10 events per cluster, there are some clusters with many events, indicating significant routing disruptions. *New-Internal-Decrease*, *Old-Internal-Increase*, *Old-Peering-Down*, and *New-Peering-Up* have relatively larger clusters than others, confirming previous findings that hot-potato changes and peering session up/down can impose significant impact [120]. Other types of causes have much smaller clusters, which is because they usually only affect individual prefixes.

Another metric to evaluate the accuracy of inferred cause is based on the number of conflicts introduced by the cause, as shown in Figure 4.7. According to §4.5, only six types of cause may have conflicts. Overall, the number of conflicts per cluster is small compared to the number of events per cluster, indicating the inconsistencies in our traces introduced by incorrect mappings or difference in probing time are rare.

We use the confidence metric introduced in the previous section to assess the likelihood of causes. Figure 4.8 shows different types of causes have different distributions of confidence value. For example, *Old-External-Worsen*, *New-External-Improve*, *Old-Lpref-Decrease*, and *New-Lpref-Increase* generally have much lower confidence values as they affects only individual prefixes. Thus we need to set appropriate thresholds to filter out different types of causes with low confidence. Throughout the rest of this section, we use a confidence value of 30 for reporting hot-potato changes (*Old-Internal-Increase* and *New-Internal-Decrease*) and 150 for reporting peering session changes (*Old-Peering-Down* and *New-Peering-Up*). Lower confidence value increases the likelihood of false positives, e.g.misinterpreting multiple simultaneous prefix withdraws from a peering as an *Old-Peering-Down*. These two values filter out 92% of the hot-potato changes and 99% of the peering session changes inferred by our algorithm. In the next subsection, we will evaluate the impact of the confidence value on our inference accuracy. We do not set any threshold for other types of causes since most of them have only one event in each cluster.

4.7.2 Validation with BGP-based inference for a Tier-1 ISP

Most previous work on diagnosing routing disruptions relies on BGP data. The closest one to ours is by Wu *et al.* [125] using BGP updates from all the border routers to peers to identify important routing disruptions. To directly compare with their approach, we implemented their algorithm, called *Wu* for convenience. We collected data via eBGP sessions to 29 border routers in a Tier-1 ISP. Note that *Wu* requires BGP data from all the border routers and it focuses on peer routes only. Given the lack of such complete data, causes reported by *Wu* on our data may be inaccurate accounting for possible mismatches.

Note that incomplete data set will only cause inaccuracies in *Wu* in terms of falsely categorizing external events to internal events or inaccurate types of internal events. The accuracy of external category is not affected by the incomplete input.

We briefly summarize *Wu*'s algorithm and our comparison methodology. *Wu* first groups a routing event from one border router's perspective into five types: no change, internal path change (using iBGP routes with nexthop changes), loss of egress point (changing from eBGP to iBGP route), gain of egress point (changing from iBGP to eBGP route), and external path change (both using eBGP route with nexthop changes). This step is accurate even with incomplete data. By correlating events from individual routers, *Wu*'s algorithm generates a vector of events for each destination prefix. The types of changes include: *transient disruption*, *internal disruption* (all routers experience internal path changes), *single external disruption* (only one router has either loss/gain of egress or external changes), *multiple external disruption* (multiple routers have either loss/gain of egress or external changes), and *loss/gain of reachability* (every router experiences loss/gain of egress). This step may introduce inaccuracy due to data incompleteness.

Compared with *Wu*'s vector change report, we first perform event based validation. We map each of our events (per source-destination based routing change) to the corresponding event in *Wu*, the prefix of which covers our destination. Each event is associated with one cause from our algorithm and one vector change type in *Wu* described above. Note that the set of causes and the set of vector change type do not have direct one-to-one mapping. To conduct a fair comparison, we combine our causes into two big categories:

Internal includes *New-Internal-Decrease, Old-Internal-Increase, Old-Lpref-Decrease, New-Lpref-Increase*, which should match *Wu's internal disruption.*

Root cause	Internal disruption	Single external	Multiple external	Loss/gain of reachability
Inte-	***34914***	5947	4494	10
-rnal	***(76.9%)***	(13.1%)	(9.9%)	(0.02%)
Exte-	16344	***44948***	***6538***	391
-rnal	(24.2%)	***(65.9%)***	***(9.6%)***	(0.6%)

Table 4.7: Event based validation: with a Tier-1 ISP's BGP data (0.29% of prefixes, 23 days).

External includes *Old-External-Worsen, New-External-Improve, Old-Peering-Down, New-Peering-Up*, which should match *Wu's single/multiple external disruption.*

These two aggregated categories are of interest because our main goal is to distinguish internal disruptions from external ones. The cause *External-AS-Change* does not have any corresponding type in *Wu*, which are omitted from comparison. Similarly, we do not compare the Same-Ingress-Same-Egress events with only internal PoP path changes as they are ignored by *Wu.*

As shown in Table 4.7, each column is the vector change type in *Wu*, while each row shows our aggregated categories. For each routing event, we identify the type y inferred from Wu as well as the category x inferred by our system. By comparing them, we generate the percentage in the table row x column y, the fraction of events in our aggregated category x that is categorized as type y in *Wu.* Bold italic font means valid matches. 76.9% of our internal events match *Wu's internal disruption*, while 75.5% of our external events match *Wu's single/multiple external disruption.*

The third step in *Wu* is to group together event vectors of different destinations belonging to the same type and transition trend. There are two types of clusters reported in the third step: hot-potato changes and peering session resets. For each of

Target AS	Hot potato			Session reset		
	Wu	Our	Both	*Wu*	Our	Both
Tier-1 ISP	147	185	101 (68%,55%)	9	15	6 (66%,40%)
Abilene (AS11537)	39	52	32 (82%,62%)	2	6	2 (100%,33%)

Table 4.8: Validation for two important clusters ($conf_{hP}$=30, $conf_s$=150)

the causes reported by us, we examine if it is also reported by *Wu*. To be more specific, for each *New-Internal-Decrease* and *Old-Internal-Increase*, we search for corresponding hot-potato change reported within that probing interval. Each *Old-Peering-Down* and *New-Peering-Up* is mapped to *Wu's peering session reset* in the same probing interval associated with the same pair of egress and neighbor AS.

The comparison for two important clusters is shown in Table 4.8. We use the confidence value of 30 for hot-potato changes and 150 for session resets based on their distinct confidence distributions shown in the previous section. The two algorithms reported 101 common hot-potato changes and 6 common session resets. Given that our system does not rely on any ISP proprietary data, it is very encouraging that we can correctly diagnose a reasonably large fraction of significant routing disruptions (68% of hot-potato changes and 66% of session resets).

We study the impact of confidence value on our inference accuracy of hot-potato changes in Figure 4.9. Unsurprisingly as we use greater confidence value, the false positive rate decreases while the false negative rate increases. With a confidence threshold of 30, we attain a balance between false positives (45%) and false negatives (32%). For session reset, the false positive and false negative rates are 60% and 34% respectively with a confidence threshold of 150.

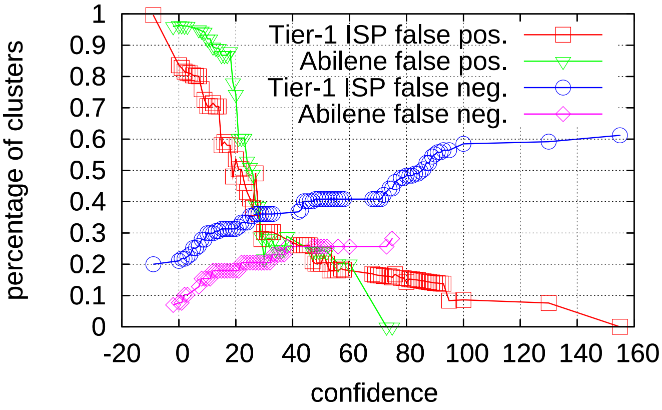


Figure 4.9: Matching rate for hot-potato changes – a common type of routing disruption.

Root cause	Internal disruption	Single external	Multiple external	Loss/gain of reachability
Inte- -rnal	***3798*** ***(78.5%)***	535 (10.1%)	555 (11.4%)	0%
Exte- -rnal	1715 (8.8%)	***16561*** ***(85.0%)***	***1208*** ***(6.2%)***	74 (0.3%)

Table 4.9: Event based validation: with Abilene BGP data (6% of prefixes, 8 days).

4.7.3 Validation with BGP-based inference and Syslog analysis for Abilene

We also validate our inference results with those of *Wu*'s algorithm executed on the BGP data from all 11 border routers of the Abilene network [6]. This provides a more complete view of routing changes for the entire network compared to the Tier-1 ISP case. Besides BGP data, router Syslog messages are also available [6] from all the Abilene border routers. Syslog reports error messages such as link down events due to hardware failure or maintenance. We can thus compare inferred link up/down causes directly with Syslog messages.

Table 4.9 shows the routing event comparison between *Wu* and our system. The matching rate for Abilene is better compared to the Tier-1 ISP case, possibly due to *Wu* improved accuracy with full visibility. The comparison for the two important clusters is shown in Table 4.8. From the Abilene Syslog, the two session resets were caused by two peering link down events which lasted for more than fifteen minutes, possibly due to maintenance. Overall, we correctly inferred 82% of the hot-potato changes and 100% of the session resets. The false positive rates are 38% for hot-potato changes and 67% for session resets respectively. 8.8% external events are mis-classified to be internal events, i.e.the accuracy of classifying external event is 91.2%. Overall, the high false positive rate is due to the lack of coverage. Since there are certain paths only traversing by a few vantage points, the evidences in the conflict graph could be few, thus the greedy algorithm is more likely to select some wrong causes. The high false positives could be further reduced by increasing the vantage points and increasing the confidence level threshold.

4.7.4 Validation with NANOG mailing list

Given that operators today often use the NANOG (North American Network Operators Group) mailing list [8] to troubleshoot network problems, we study the archives of the mailing list messages over the time period of our study. All together we analyzed 2,694 emails using keyword searches and identified six significant routing disruptions with details described below. One interesting observation is that even when we did not directly probe the problematic AS discussed in the email as the target AS, we are still able to identify the following four disruptions due to their wide-spread impact:

1. Apr. 25, 2007, between 19:40 to 21:20 EDT, NANOG reported a Tier-1 ISP Cogent (AS174) experienced serious problem on its peering links causing many withdrawals. The target AS during this time was AS3320. Our system observed increased number of routing events: 120 detected events were clustered into 96 causes of *External-AS-Change*, affecting 7 sources and 118 destinations. 87 of the events were associated with 42 destinations which were Cogent's customers. They all switched away from the routes traversing Cogent. Significant delay increase was also observed.

2. May 21, 2007, around 21:50 EDT, NANOG reported a backbone link fiber cut between Portland and Seattle in the Level3 network (AS3356), resulting in reachability problems from Level3's customers. The target AS at that time was also AS3320. Our system detected 45 events clustered into 36 causes of *Old-External-Worsen*, affecting 5 probing sources and 12 destinations. They all switched from routes traversing Level3 to those traversing AS3491 in the Seattle PoP.

3. Jun. 14, 2007, NANOG reported a core router outage around 6am EDT in the Qwest network (AS209), affecting the performance of several networks and their customers. The target AS studied at the time was AS3561. Our system reported 24 events clustered into 23 causes of *External-AS-Change* switching from paths traversing AS209 to those traversing AT&T (AS701, AS703) around the outage time, affecting 6 probing sources and 24 probing destinations.

4. Sep. 19, 2007, 13:00 EDT, NANOG reported that 25 routers in the Broadwing network (AS6395) had a misconfiguration resulting in BGP session removal. It caused multiple single-homed customers disconnected from Internet. Immediately after that, our system detected 81 events clustered into 64 causes of *Old-External-Worsen*, for 76 destinations from 10 sources. The target AS, AS2914, switched from the old routes traversing Level3 (AS3356) and Broadwing to new routes traversing other peers, e.g.AS209 and AS7018.

We missed two NANOG-reported events related to routing and performance disruptions during our study. The first was on May 16, 2007, from 13:10 to 14:20 EDT, related to a hardware problem on the peering link between AT&T and Broadwing in Dallas. Our system did not capture any routing changes during this time period at that location. The second event was on May 30, 2007, around 13:00 EDT, related to significant performance degradation, along with temporary loss of reachability from Sprint in the Pittsburgh area, as confirmed from Sprint. The target AS probed was AS3561. Although our system did not report routing changes related to Sprint, it did observe abnormal incomplete traces from PlanetLab hosts in Pittsburgh.

To summarize, our system may miss some localized disruptions due to limited coverage. But it is able to capture disruptions with global impact even when they are not directly caused by the target AS being probed.

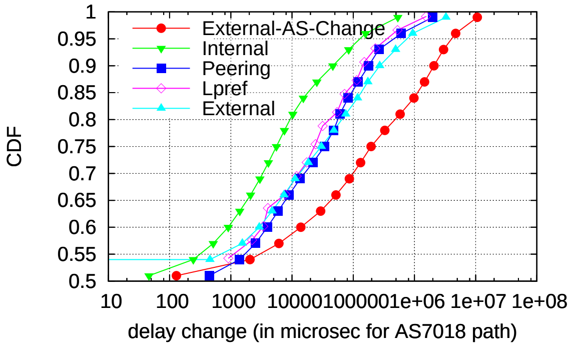


Figure 4.10: Delay change distribution in each category for AS7018 (actual path delay).

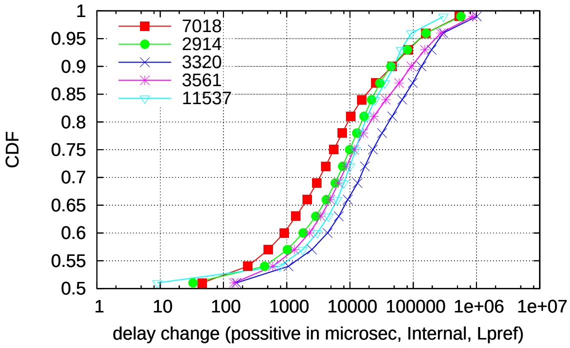


Figure 4.11: Comparison between absolute path delay and target delay changes (AS7018, type:new distance decrease and external AS path change)

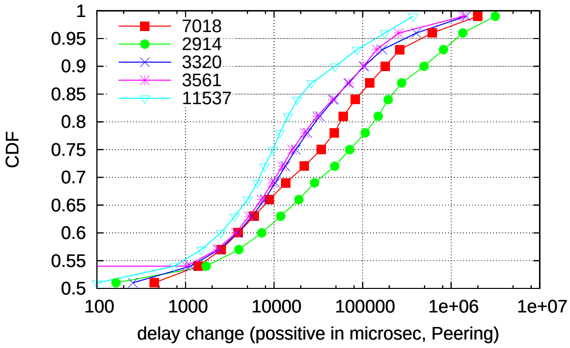


Figure 4.12: Delay change distribution of routing change across different target AS (old dist. inc, target)

4.8 Performance Impact Analysis

Routing events are known to introduce disruption to network path performance. Unlike the past work that relies on routing feeds to diagnose routing events, end-host

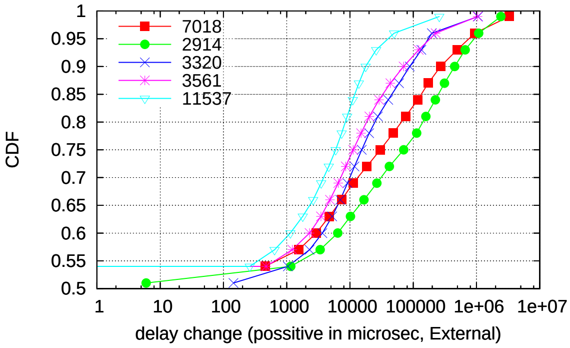


Figure 4.13: Delay change distribution of routing change across different target AS (old edge down, target)

probing used in our system enables us to understand the impact of routing events on path performance. In this section, we study to what extent end-to-end latency is affected by different types of routing events and variation cross different ISPs.

Figure 4.11, 4.12, and 4.13 illustrates the latency change for different routing events in AS7018. For clarity, we only show five types of events: *Internal* (*Old-Internal-Increase*, *New-Internal-Decrease*), *Peering* (*Old-Peering-Down*, *New-Peering-Up*), *Lpref* (*Old-Lpref-Decrease*, *New-Lpref-Increase*), *External* (*Old-External-Worsen*, *New-External-Improve*), and *External-AS-Change*. Because we use log scale on the y-axis, the graph does not show the cases where latency change is negative. Given that almost all the curves start from 0.5, it implies latency has the same likelihood to improve or worsen after these events. A noteworthy observation is external events (*External-AS-Change*, *External*, and *Peering*) have much more severe impact than internal events (*Internal*), suggesting that AT&T's network is engineered well internally. We observe similar patterns for the other ISPs studied.

Figure 4.11 illustrates how the latency change induced by the same event type varies across different ISPs. We omit *External-AS-Change* here because this type is not directly related to a target ISP. Figure 4.11 shows little difference among the five target ISPs in terms of latency change caused by internal events, as most changes are relatively small. Turning to Figure 4.12 and 4.13, the difference between the ISPs becomes much more noticeable. AS11537 appears most resilient to external events in terms of latency deterioration while AS2914 appears worst. The relative difference between the ISPs is consistent in both graphs, suggesting that customers sensitive to performance disruptions should take great care in selecting the appropriate providers.

4.9 System Evaluation

In this section, we show that our system imposes a small amount of memory and CPU overhead to perform event identification, classification, and inference. We evaluate our system on a commodity server box with eight 3.2GHz Xeon processors and 4 GB memory running Linux 2.6.20 SMP.

Memory usage The memory usage of our system is comprised of (i) the two most recent routing states of the target ISP extracted from the traces, and (ii) the evidence and the conflict graphs constructed from the two routing states (see §4.3). The first type of memory usage is relatively static over time since the overall topology and routing of a target ISP do not change frequently. The memory usage of the two graphs is more dynamic and depends on the number of detected routing events. The former has been dominant throughout our evaluation period of 111 days, because the

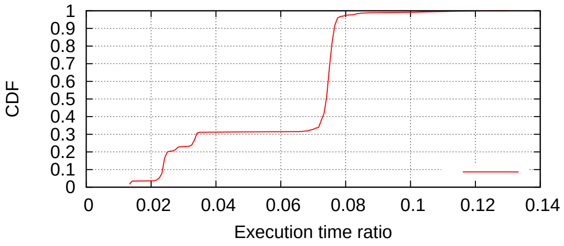


Figure 4.14: Relative execution time compared with the probing interval

number of traces always outweighs the number of routing events. The total memory footprint of our system never exceeds 40 MB.

Execution speed We now evaluate whether our system can keep up with the continuously incoming routing states. Running our system in real time allows us to quickly detect and raise alert on significant routing disruptions to take corrective actions promptly.in a timely manner. Figure 4.14 plots CDF of the ratio of the execution time relative to the routing state interval. It is clear that the execution time is much shorter, all of them are within one eighth of the routing state interval.

4.10 Summary

Given our increasing dependence on the Internet for QoS-sensitive applications, it is critical to effectively diagnose routing-induced performance problems. Past work on diagnosing routing events has relied on routing feeds from each ISP. These techniques have fundamental limitation due to the limited coverage of routing data, as discussed in Chapter III. Motivated by these analysis, we aim to develop new techniques for diagnosing routing events from end systems.

In this chapter, we have presented the first system to accurately and scalably diagnose routing disruptions purely from end-systems without access to any sensitive data such as BGP feed or router configurations from ISP networks. Our approach differs from recent work in that it purely relies on probing launched from end-hosts and does not require any ISP proprietary information. We summarize our key techniques and results in the following.

To diagnose the routing changes from a given ISP, our system first learns the continuous routing state of the ISP using end-host based continuous probing. Based on the change in routing state, it identifies and classifies individual routing events. Because a single routing disruption often introduces many routing events, our system applies an inference algorithm to find explanation for cluster of events occurring closely in time. The inference algorithm is a simple greedy algorithm on two bipartite graphs representing observed routing events, possible causes, and the constraints between them. Our system effectively infers the most likely causes for routing events detected through light-weight traceroute probes. It then uses the latency measurements in the probes to quantify the impact of these routing events. We comprehensively validate the accuracy of our results by comparing with existing ISP-centric method, publicly-available router configurations, and network operators' mailing list.

To summarize, our work is the first to enable end systems to scalably and accurately diagnose causes for routing events associated with large ISPs without requiring access to any proprietary data such as real-time routing feeds from many routers inside an ISP. The system in this chapter focuses on diagnosing routing-induced disruptions. Routing change is only one type of causes resulting in performance degradations.

Unexpected increase of traffic volume may lead to congestion on the bottleneck links causing disruptions to the application. Even with over-provisioned capacity on most links, ISPs may still employ various prioritization techniques to shape the traffic differently to avoid unexpected performance degradation. Various traffic prioritization technologies are available on today's commercial routers such as queuing mechanisms. Different types of traffic may experience different performance within the same ISP network due to various reasons such as commercial relationship. In chapter V, we examine another type of performance problems induced by ISP policies. We aim at detecting the problem of detecting traffic differentiation in backbone ISPs. Similar to routing data, most ISPs do not reveal the details of their network policies and configurations. In next chapter, we aim to develop an end-host based system that can detect traffic differentiation without any ISP cooperation.

CHAPTER V

Detecting Traffic Differentiation in Backbone ISPs

5.1 Introduction

There is significant controversy surrounding the topic of network neutrality on the Internet. Since its early days, Internet is designed under the end-to-end principle which argues for intelligent end systems and a “dumb” network. Under this principle, network should deliver traffic with best effort and should not treat traffic preferentially based on various properties such as IP address, port number, or packet content. In recent years, a variety of new applications have emerged and proliferated on the Internet. Some require high bandwidth (e.g.peer-to-peer file sharing and video streaming) while others require low latency and loss rate (e.g.voice-over-IP and online gaming). Such trend has inspired ISPs to perform various types of traffic shaping to limit network resource usage and introduce tiered services to raise profit.

Residential broadband ISPs, such as Comcast, are known to treat traffic differently, e.g.by limiting the bandwidth usage of peer-to-peer file sharing applications.

Cellular network carriers, such as AT&T, have also been reported to restrict the usage of video streaming services to preserve their limited wireless spectrum [2]. Researchers have proposed various techniques for detecting traffic differentiation. Beverly *et al.* presented one of the first measurement studies of port blocking behavior from the edge of the Internet [27]. POPI is another tool for determining router traffic differentiation policy based on port numbers via end-host measurements [78]. More recently, Dischinger *et al.* developed tests for detecting whether broadband ISPs rate-limit or block BitTorrent traffic [45]. Besides these active measurement techniques, Tariq *et al.* proposed to identify differentiation by applying statistical method to passive measurements from end hosts [118]. Yet so far, there has been no detailed and comprehensive study on the current practice of traffic differentiation inside the Internet core. Traffic differentiation in the core arguably has a much wider scope of impact, as such policies affect much more traffic compared to the policies near the edge of the Internet.

In this chapter, we consider the problem of detecting traffic differentiation in backbone ISPs. Different types of traffic may experience different performance within the same ISP network due to various reasons. An ISP may throttle the traffic from a neighbor (e.g.a free peer) by routing the traffic over a low-capacity link. It may also prevent the traffic of an application (e.g.BitTorrent) from disrupting other traffic via weighted fair queueing. The ability to detect traffic differentiation enables customers to develop the appropriate strategies for improving their application performance. For instance, large content providers, like Microsoft, Google, and Yahoo, strive to ensure their Internet applications outperform those offered by their competitors. If a content provider knows the average loss rate of its traffic traversing a particular ISP is twice that of its competitor, it may choose to negotiate better service level agreements (SLA) with that ISP. Small customers will also benefit from such differentiation information. For instance, they may change port numbers or encrypt packets to circumvent content-based differentiation employed by their ISP [21].

Most ISPs do not reveal the details of their network policies and configurations. Realizing this problem, we aim to develop an end-host based system that can detect traffic differentiation without any ISP cooperation. Such a system is not only easily deployable but also applicable to many different ISPs. To build such a system, we face two key challenges: i) unlike in the case of broadband ISPs, most end hosts are not directly connected to backbone ISPs. We need to intelligently select probing destinations to cover the relevant internal paths of backbone ISPs while complying with the requirement of limited network and CPU resources on end hosts; ii) measurement data taken from end host is susceptible to various types of noise on the host or in the network. We need to ensure our detection results are not distorted by noise.

NVLens is the first fully operational system that can detect traffic differentiation in backbone ISPs by accurately and scalably monitoring packet loss behavior. It relies on an intelligent path selection scheme to detect both content- and routing-based differentiation while systematically balancing path coverage and probing overhead. It leverages statistcal hypothesis tests to identify significant loss rate differences between different types of traffic measured along the same ISP internal paths after discounting the effects of measurement noise. Furthermore, it uses a novel technique for cross-validating the statistical test results and the Type-of-Service (TOS) value set by ISPs.

Type	Examples
Packet headers	source/destination port numbers, protocol type
Application layer info	application protocol headers (e.g.HTTP header, BitTorrent header), application payload
Traffic behavior	flow rate, flow duration, packet size, packet interval
Routing info	previous-hop AS, next-hop AS, source/destination IP addresses
Available resources	queue length, link utilization, router load and memory

Table 5.1: Information commonly used to determine policies for differentiation.

By studying 18 large ISPs spanning 3 major continents over a period of 10 weeks, NVLens provides concrete evidence of traffic differentiation based on application types and neighbor ASes. We identified 4 ISPs that exhibit large degree of differentiation on VoIP, BitTorrent, PPLive, and SMTP traffic compared to HTTP traffic. We also identified 10 ISPs that treat traffic differently based on its previous-hop ASes, reflecting different business contracts. The significance of differentiation increases with network load, suggesting that differentiation is likely to be triggered by resource competition. The absolute loss rate difference between certain pairs of applications or previous-hop ASes can exceed 5%, large enough to impair the performance of many TCP-based applications. Interestingly, we find a few ISPs simply rely on port numbers to perform traffic differentiation irrespective of actual payload. We further validate our detection results on paths where we have two-ended control.

5.2 Traffic differentiation

An ISP may use various information in traffic and routers to construct differentiation policies. Table 5.1 enumerates a list of such potential factors [137]. First, an ISP may provide differentiated services based on the application type for security or business reasons. It is well-known that broadband ISPs drop certain SMTP traffic to fight spams and throttle BitTorrent traffic to restrict bandwidth usage. Application types can be determined from packet header fields or application layer information [88]. Even with encrypted traffic, there are sophisticated techniques that can infer application types by identifying certain traffic behavior [124]. Second, an ISP can differentiate traffic according to routing information, reflecting distinct business contracts with its customers and peers. An ISP may favor inbound traffic from customers who pay for premium services or disfavor inbound traffic from peers. This type of differentiation can be applied based on the previous-hop/next-hop ASes or the source/destination IP addresses. Such information can be easily extracted from packet headers and routing state. Third, an ISP may enforce differentiation policies according to available resources. Using the link utilization information readily available from SNMP [38], it may slow down traffic with low priority to preserve sufficient bandwidth for other traffic.

It is feasible to implement traffic differentiation in a backbone network with many high-speed links. Today's router already has support for various queuing mechanisms to fulfill the need of traffic engineering, quality of service, and security guarantees. Figure 5.1 illustrates a common architecture for implementing differentiation within

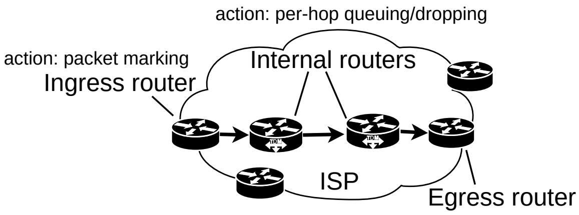


Figure 5.1: An example of differentiation implementation.

a backbone ISP [137]. The ingress border routers perform traffic classification by marking packets according to packet header fields and routing information, such as port numbers and previous-hop/next-hop ASes. The marking is usually applied to the Type-of-Service (TOS) field in the IP header. The internal routers perform traffic shaping according to the TOS value in the packets [68]. There are various queuing and dropping mechanisms that provide different levels of service to traffic, e.g.priority queuing, proportional share scheduling, and policing [37]. These mechanisms differ in details of how and when differentiation is carried out. In §5.6.7, we demonstrate traffic differentiation can be easily implemented on today's commercial routers in testbed experiments.

Other than the router marking-based mechanisms using packet header information, ISPs may perform deep packet inspection (DPI) [46] to classify application types according to packet content. Some DPI devices can perform pattern matching in packet payload with hardware support for 100 Gps links [1, 40]. Because DPI devices can be quite expensive, they are usually deployed only at selected locations.

In this work, we examine all types of differentiation listed in Table 5.1 except for the one based on traffic behavior (Table 5.1 row 4) due to limitations of end-host based probing. In fact, behavior-based differentiation is very expensive to implement

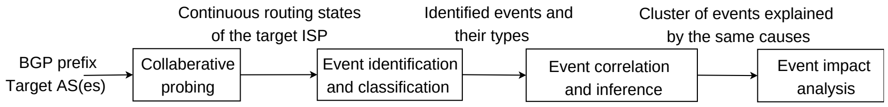

Continuous routing states of the target ISP
Identified events and their types
Cluster of events explained by the same causes
BGP prefix
Target AS(es)
Collaberative probing
Event identification and classification
Event correlation and inference
Event impact analysis

Figure 5.2: Select path to discover various types of traffic differentiation.

by ISPs due to the required per-flow state information and potentially high false positives. Our goal of detecting these four types of differentiation guides the design of path selection and probe packet composition in NVLens. By providing concrete evidence of differentiation, we hope to stimulate more research to fully understand possible differentiation policies in backbone ISPs. We plan to extend NVLens to detect other differentiation polices once they are known.

5.3 Methodology

NVLens detects traffic differentiation inside a particular ISP by launching probes from a distributed set of end systems. For this purpose, we have to decide what paths to measure, how to measure each path, and how to identify differentiation based on measurement results. We address these three issues below.

5.3.1 Path selection

NVLens is designed to detect traffic differentiation based on packet headers, application layer information, and routing information (described in Table 5.1). Figure 5.2 illustrates how NVLens uses measurements from end systems to identify differentiation by a particular *ISP W* [137]. In the leftmost figure, an end host probes three paths to different destinations, sharing the same ingress and egress within *ISP W*, but diverging into three distinct next-hop ASes after leaving the egress. By comparing the internal performance of the three paths between the ingress and egress, NVLens can determine whether *ISP W* treats traffic differently based on the next-hop ASes or destinations. Similarly, the middle figure shows how NVLens detects differentiation based on previous-hop ASes or sources. In the rightmost figure, an end host probes a path that traverses the same ingress and egress of *ISP W* to the same destination. By comparing the internal path performance measured by packets of different applications (e.g.HTTP *vs.* BitTorrent), NVLens can detect differentiation based on content, such as packet headers and application layer information. We leave the discussion of resource-based differentiation to §5.6.5.

To detect traffic differentiation inside a particular ISP, we must devise an intelligent path selection strategy to ensure good coverage and low overhead. On the one hand, a backbone ISP typically consists of multiple PoPs (Points of Presence) at many geographic locations. We want to cover as many distinct PoP pairs as possible in order to quantify the scope of traffic differentiation policies inside the ISP. On the other hand, NVLens relies on end hosts to perform measurements. While this makes NVLens easily deployable and applicable to different ISPs, we must aggressively reduce the measurement overhead to comply with the requirement of limited CPU and network resources at each host.

Given a target ISP, a list of probing sources, and all the destination prefixes on the Internet, a naive approach is to probe all the prefixes from all the sources. This may lead to both wasteful probes that do not traverse the target ISP and redundant probes that traverse the same internal paths multiple times. To avoid these two problems, we frame the path selection problem as follows.

1. Each three-tuple $(src, ingress, egress)$ is traversed at least R times by probes to different destinations.

2. Each three-tuple $(ingress, egress, dst)$ is traversed at least R times by probes from different sources;

3. A probing source does not send more than m probes.

Here, src is a probing source, dst is a destination prefix, and $ingress$ and $egress$ are the PoPs in the ISP.

Conditions 1 and 2 allow us to detect differentiation based on routing information, i.e.previous-hop and next-hop ASes respectively. We can also detect content-based differentiation by probing the same path with packets of different applications. R is a tunable redundancy factor that determines the tradeoff between probing overhead and coverage. A larger R will increase not only the chance of detecting routing-based differentiation but also the amount of probing traffic. Condition 3 caps probing load at each source.

This problem is an instance of the set covering/packing problem [71, 83]: given multiple sets over a universe of elements, pick a subset of input sets such that each element is included at least R times (covering constraint), and no element is included more than m times (packing constraint). In our case, the input sets are the probes between source-destination pairs, and the elements are probers and the three-tuples of $(src, ingress, egress)$ and $(ingress, egress, dst)$. A probe typically contains all three element types. This formulation enables us to perform both redundancy elimination and probing load assignment systematically. While this problem is NP-hard, we use a greedy based approximation: at each step, we select the probe that covers the most uncovered elements without exceeding the probing threshold m. This process continues until all the elements are covered at least R times. R is a pre-defined parameter called redundancy factor.

5.3.2 Loss rate measurement

NVLens focuses on detecting traffic differentiation that affects performance. The effect of this type of differentiation is more stealthy, compared with other brute-force differentiation schemes used by broadband ISPs, such as traffic blocking and TCP SYN/RST [45]. Currently, NVLens measures loss rate, one of the most important performance metrics. It can also be extended to measure other metrics, e.g.latency, jitter, and reachability.

Given a path, NVLens measures the loss rate as follows. First, to reduce probing overhead, NVLens only probes the hops that map to an ingress or an egress of a target ISP instead of all the hops along the path, given that we are only interested in detecting differentiation inside the ISP. Second, to measure the loss rate to a particular hop, NVLens sends probe packets with pre-computed TTL (Time-to-Live) value which will trigger ICMP time exceeded response from that hop. In essence, these packets are similar to traceroute probes.

Since packet loss may occur in either direction, we use large probe packets to ensure the measured loss is mostly due to forward path loss. The assumption is that large probe packets are more likely to be dropped than small ICMP packets on the reverse path. This has been widely adopted in previous work [79, 81]. To avoid triggering ICMP rate limiting, NVLens probes each hop once per second for 200 times, allowing us to detect loss rate as small as 0.5%. Probing each hop more times increases the sensitivity of loss rate detection but also the probing overhead. We subtract the measured loss rate of the ingress from that of the egress to obtain the loss rate of the internal path. In §5.5, we conduct controlled experiments to confirm that our loss rate measurements are not distorted by reverse path loss, ICMP rate-limiting, or load on probing source.

To detect content-based differentiation, we measure loss rate of an internal path using different application traffic. We select six representative applications with distinct QoS (Quality of Service) requirements: HTTP (default port 80), BitTorrent (P2P file sharing, port 6881), SMTP (email, port 25), FTP (file transfer, port 21), PPLive (online streaming, port 4004), and VoIP (port 5060). Except for HTTP, the remaining five applications are selected based on how likely they will be treated differently by backbone ISPs. HTTP, one of the most commonly-used application, is used as the baseline to compare performance with other applications. ISPs may slow down BitTorrent, FTP, and PPLive traffic due their high volumes. Similarly, ISPs may disfavor SMTP traffic due to email spam concerns. We also test VoIP traffic because many ISPs provide their own VoIP service using port 5060, giving incentives to preferential treatment.

We construct probe packets with application-specific content captured from real application traces. This eliminates any need to understand the protocols of proprietary applications, such as PPLive or VoIP. To enable fair comparison between the loss rate of different applications, all probe packets are chosen to be of the same size.

Because NVLens relies on TTL-based probes to measure path performance, it cannot fully mimic the temporal behavior of real applications. Probing too aggressively may trigger ICMP rate-limiting on routers. The alternative of running real applications on hosts under our control is less appealing, due to access to limited number of hosts and the challenges of inferring the properties of ISP internal paths with complex tomography techniques [118].

5.3.3 Differentiation detection

NVLens detects differentiation by observing the performance difference measured along the same ISP internal path using different types of probing traffic. We must ensure that the observed differences accurately reflect how an ISP treats different types of traffic. Since it is difficult to take two loss rate samples at the same time under the same network conditions, we cannot detect differentiation solely based on the difference between two samples.

We first introduce a few notions before describing the details of our differentiation detection scheme. For a target ISP I, we define $l_{\{s,d,a,t\}}$ to be a loss rate sample measured along an internal path of ISP I from a probing source s to a destination d, using probing packets of application a at time t. We use the term *set* to denote a set of samples that satisfy certain conditions. For example, $set_{\{s,d,a\}}$ includes all the samples measured along a path from s to d using packets of application a. Similarly, $set_{\{AS_p,P_i,d,a\}}$ includes all the samples measured along the paths traversing previous-hop AS_p and ingress P_i to destination d, using packets of application a.

Our basic assumption is that the loss rate samples in a *set* follow a particular underlying distribution. We can detect differentiation by comparing the distributions of two candidate sets. We use $pair_{\{s,d,a1,a2\}}$ to denote two candidate $set_{\{s,d,a1\}}$ and $set_{\{s,d,a2\}}$. We can compare the distributions of an *application* $pair_{\{s,d,a1,a2\}}$ to detect content-based differentiation between $a1$ and $a2$. Similarly, we use $pair_{\{AS_{p1},AS_{p2},P_i,d,a\}}$ to denote two candidate $set_{\{AS_{p1},P_i,d,a\}}$ and $set_{\{AS_{p2},P_i,d,a\}}$. We can compare the distributions of an *AS* $pair_{\{AS_{p1},AS_{p2},P_i,d,a\}}$ to detect previous-hop AS based differentiation between AS_{p1} and AS_{p2} at ingress P_i. As long as the underlying distributions are stable and the two candidate sets include enough samples, the comparison result should be independent from the measurement time of individual samples.

Given a pair of sample sets, we apply statistical hypothesis tests to determine if there is significant difference between their distributions. Several commonly-used hypothesis tests exist to compute the statistical significance of difference between two input sets. Since the distribution of the loss rate samples in an input set is unknown, we choose the Kolmogorov-Smirnov (K-S) test [114] which makes no assumption about the input sample distribution. The K-S test compares the distance of the two empirical cumulative distribution functions F_1 and F_2 of the two input sets. It computes the Kolmogorov-Smirnov statistic $D_{1,2} = \sup_x |F_1(x) - F_2(x)|$, where sup is the supremum, under the null hypothesis that the two sets of samples are collected from the same distribution. The null hypothesis test is rejected at significance level α if $\sqrt{\frac{n_1 n_2}{n_1+n_2}} D_{1,2} > K_\alpha$, where n_1 and n_2 denote the sizes of the input sets, and K_α is the critical value in the K-S statistic table. Given multiple paths from one ISP,

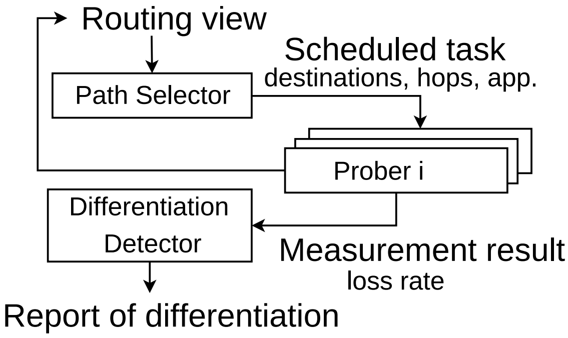


Figure 5.3: The NVLens architecture

we adjust the threshold of confidence interval differently according to the number of paths from each ISP.

To verify whether the K-S test statistic is independent from the measurement time of individual samples, we use Jackknife [121], a commonly-used non-parametric resampling method, to evaluate the stability of the K-S test statistic. The idea is to randomly select half of the samples from the two original input sets and apply the K-S test on the two new subsets of samples. This process is repeated r times. If the results of over β% of the r new K-S tests are the same as that of the original test, we conclude that the original K-S test statistic is independent from the measurement time of individual samples. We use $r = 400$, $\alpha = 95\%$, and $\beta = 95$ to ensure 95% confidence interval and up to 5% false positives. We will justify the choice of confidence interval in §5.6.

ASN	ISP	Tier	PoP	PoP-PoP	PoP-AS
209	Qwest	1	49	716	337
701	UUNet		139	2125	806
1239	Sprint		57	1498	1170
1668	AOL Transit		25	232	102
2914	Verio		46	501	351
3356	Level3		71	1750	653
3549	Global Crossing		59	677	371
3561	Savvis		38	502	195
7018	AT&T		112	822	430
2828	XO	2	45	539	176
2856	British Telecom		32	419	119
3257	Tiscali		30	267	138
3320	Deutsche Telekom		64	115	195
5511	France Telecom		23	303	82
6395	Broadwing		19	137	66
6453	Teleglobe		44	538	208
16631	Cogent		69	1787	152
6461	AboveNet	3	44	261	316

Table 5.2: 18 target ISPs: # of PoPs, # of PoP-PoP pairs, # of PoP-neighbor AS pairs.

5.4 Implementation

The implementation of NVLens is illustrated in Figure 5.3. It has three major components:

Path selector takes routing views as input and compute a task list of probing destinations for each prober. The routing views are the traceroute measurements conducted from all the probers to all the destination prefixes on the Internet. The path selector uses the routing views to learn the ingress and egress of the target ISPs that each path traverses. The routing views are updated daily to keep up with the evolution of ISP topologies. The path selector implements the greedy algorithm described in §5.3.1. Note that path selection is performed for multiple target ISPs simultaneously. This significantly reduces probing overhead by leveraging the fact that a single probe often traverses multiple target ISPs, allowing us to cover the same set of three-tuple elements (defined in §5.3.1) with fewer probes compared with probing each ISP separately.

Probers run on a distributed set of end hosts, probing all the destinations in their task list periodically. After completing each round of probing to all the destinations, the probers send their measurement results to the differentiation detector for further processing. Probings are conducted with a customized version of traceroute that probes multiple hops of a path and multiple destinations in parallel. The probe packets are constructed to reduce the probability that different probe packets from the same source to the same destination take different IP-level paths due to load-balancing [19].

Differentiation detector first filters the noise in the measurement results due to overloaded probers or reverse path losses. It then tries to detect differentiation based on content, previous-hop AS, or next-hop AS, following the process described in §5.3.3. Finally, it performs detailed analysis on differentiation policies, such as what input information they use, whether they are affected by network load, and how significant their impact is.

We deployed NVLens on the PlanetLab testbed [101]. It uses all the PlanetLab hosts across about 300 distinct sites. Each round of probing takes roughly two hours to complete. The results are based on 74 days of data collected during a period between

August 2008 and October 2008. Each *set* includes around 1,000 loss rate samples. We run multiple instances of NVLens to take measurements of the six applications described in §5.3.2 in parallel. We randomize the order of destinations to probe in each round to reduce the chance of a path being simultaneously measured by multiple instances. We studied 18 large ISPs covering major continents including North America, Europe, and Australia, consisting of 9 Tier-1 ISPs, 8 Tier-2 ISPs, and 1 Tier-3 ISP. Table 5.2 shows NVLens has a decent coverage of internal paths and interconnections, traversing 115-2125 ingress-egress pairs and 66-1170 PoP-AS pairs for each ISP. A PoP-AS pair represents an interconnection between a neighbor AS and the target ISP at the corresponding PoP. We map an IP address to a PoP using a name rule set derived from Rocketfuel [110].

5.5 Eliminating noise effects

Loss rate measurements taken by end-hosts are susceptible to various types of noise on the host and in the network. As mentioned in §5.3.2, the inaccuracy of loss rate measurements is likely to be caused by three main factors: 1) overloaded prober; 2) ICMP rate limiting at router; and 3) loss on reverse paths. In this section, we investigate the effect of these three factors and develop ways to mitigate their impact. Occasionally, our measurements may be disturbed by routing events. We simply discard the samples during the time period when routing events are detected by observing path changes between two consecutive probes.

Many ISPs perform load balancing using equal-cost multi-paths (ECMP) to ensure effective utilization of network resources [18]. Load balancing is usually performed based on the five tuple $(srcip, dstip, srcport, dstport, proto)$. Thus, different application packets, e.g.BitTorrent and HTTP, may take different internal IP-level paths between the same ingress and egress, given their different destination ports (e.g.6881 *vs.* 80). We carefully design experiments to ensure our differentiation detection is not affected by potential performance difference of ECMP paths.

5.5.1 Overloaded probers

Previous work has shown measurement inaccuracies caused by resource contention on probing hosts in PlanetLab experiments [108]. To deal with this problem, we continuously monitor the resource usage on each prober. We compute per minute average CPU utilization on each prober using three instantaneous load samples obtained by running the `top` command. We can then investigate the relationship between CPU utilization and measured loss rate by temporally correlating these two types of samples. This allows us to identify abnormal loss rate samples collected in period when high CPU utilization causes heavy losses on prober.

To determine an appropriate cut-off threshold of high CPU utilization, we design the following controlled experiment to study the effects of CPU utilization on loss rate measurements. We select a pair of lightly-loaded PlanetLab machines at the same site. One machine acts as a "prober" to transmit one 1000-byte probe packet per second. The other machine acts as an "acker" to receive probe packets and return 40-byte ACKs. In essence, the "prober" behaves just like a real NVLens prober that measures loss rate. We then use a program to gradually increase the CPU utilization on the "prober" while keeping the acker lightly loaded.

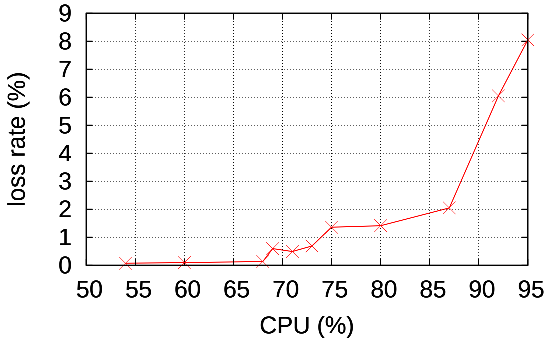


Figure 5.4: Impact of CPU utilization on loss rate.

Figure 5.4 illustrates the relationship between CPU utilization and loss rate measured by the "prober". Because loss is unlikely to occur on the light-loaded acker or on the local area network between the "prober" and the acker, the measured loss rate is almost certainly due to the CPU load on the "prober." Clearly, the loss rate jumps up when the CPU utilization reaches above 65%. We repeat this experiment on ten pairs of PlanetLab hosts across different sites and consistently find 65% to be a good cut-off threshold. In our data, 15% of the samples are discarded by applying this threshold.

5.5.2 ICMP rate limiting

ICMP rate limiting is often configured on a per-router basis to prevent router overload. If triggered, it may significantly inflate the measured loss rate. To prevent this, we deliberately keep a large probing interval, e.g.only one probe packet is sent on a given path per second. We use the following experiments to confirm that this probing interval is large enough to avoid triggering ICMP rate limiting.

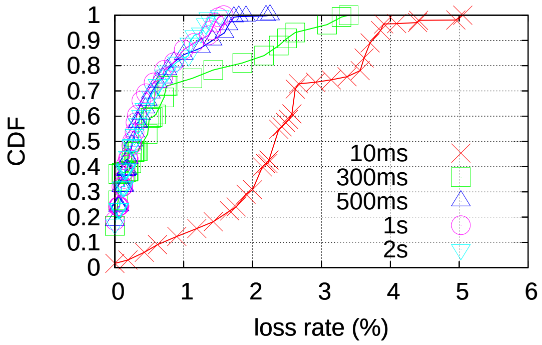


Figure 5.5: Impact of probing frequency on loss rate.

We conducted five sets of experiments by measuring the loss rate of all the internal paths of the 18 target ISPs from all the probers. We gradually increase the probing interval for each set of experiments from 10ms to 2s. The smaller the interval is, the more likely a router along a path may rate-limit the ICMP time-exceeded replies. As shown in Figure 5.5, the measured loss rates on 30% of the paths increase significantly when the probing interval changes from 500ms to 300ms. This indicates the rate-limiting threshold of the routers on those paths is between 300ms and 500ms. The loss rates measured using the intervals of 1s, 2s, and 500ms are very close, suggesting that the 1-second probing interval is sufficiently large.

5.5.3 Loss on reverse path

NVLens relies on single-ended probes to measure loss rate. The measured loss rate can be inflated due to reverse path loss. Since large packets are more likely to be dropped [79], we use 1000-byte probe packets to ensure the measured loss is mostly on forward paths. We study the effect of packet size on measured loss rate using controlled experiments. We conducted three sets of experiments by measuring the

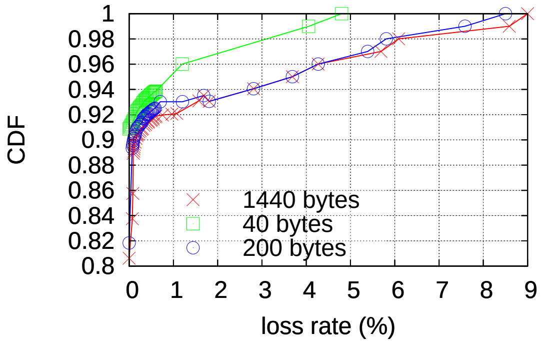


Figure 5.6: Impact of probing packet size on loss rate.

loss rate of all the ISP internal paths using probe packets of 40 bytes, 200 bytes, and 1440 bytes. As shown in Figure 5.6, the measured loss rate increases with probe packet size. Since the size of the ICMP responses is the same, this confirms that bigger probe packets are more likely to encounter losses on forward path. Nonetheless, the loss rates measured by 200-byte and 1440-byte packets are roughly the same, suggesting the effects of packet size on forward path loss diminish when packet size exceeds 200-byte.

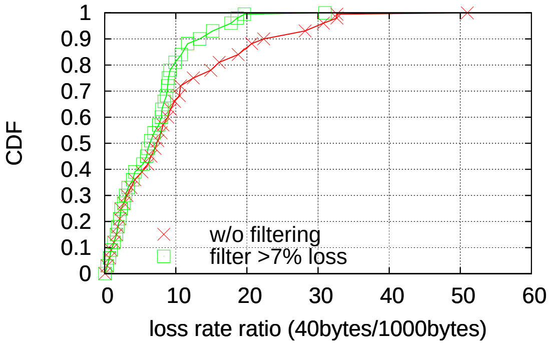


Figure 5.7: Loss rate ratio (filtering vs. no filtering).

The loss rate measured by 40-byte probe packets is much smaller. Since the ICMP response packets are of similar size (usually 56 bytes), we can use this loss rate as the upper bound of the loss rate on reverse path. To further reduce the impact of reverse path loss, we compute the ratio of the loss rate measured by 40-byte packets and that by 1000-byte packets. This loss rate ratio is an estimate of the relative error of loss rate measurements induced by reverse path loss. From Figure 5.7, we observe that the loss rate ratio is less than 10% for 70% of the paths. We find the loss rate ratio tends to be high when measured loss rate is high. By filtering out 6% of the samples with abnormally high loss rate ($\geq$ 7%), the loss rate ratio is within 10% for 80% of the remaining paths.

5.5.4 Effects of load balancing

Load balancing is observed extensively in our measurements, e.g.BitTorrent traffic and HTTP traffic take different internal IP-level paths between 48% of the source-destination pairs. To eliminate the effect of load balancing, we take a conservative approach in detecting content-based differentiation. We first detect potential differentiation for each application *pair* from the initial measurement data. We then verify that the detected differentiation still exists when the probe packets of the two applications traverses the same internal IP-level path. Since load balancing algorithms usually use both source and destination ports to choose an internal path, we fix the source port of one application while changing the source port of the other application until the probe packets of both applications follow the same internal IP-level path. The results in §5.6 are obtained after applying this controlled procedure to each application pair.

App	ISP	Paths	δ_{tos}	TOS_δ	Same	$\frac{1}{3}$
BT	Tiscali	3794,19	100	99	12,0.06	3579,18
PPLive	Tiscali	825,4.1	100	85	24,0.1	903,4.4
VOIP	UUNet	172,3.2	100	68	11,0.2	157,2.9
VOIP	Sprint	203,2.1	100	96	25,0.2	237,2.5
SMTP	UUNet	573,11	100	93	9,0.02	595,11.4
SMTP	Verio	388,7.2	100	97	52,0.9	401,7.4

Table 5.3: K-S test results for content-based differentiation.

5.6 Experimental results

In this section, we provide concrete evidence of traffic differentiation based on content and routing in backbone ISPs. We study the types of information used to construct content-based differentiation policies and the impact of business relationship on routing-based differentiation policies. Without access to ISPs' proprietary policy configurations, we leverage both TOS value in probe packets and two-ended controlled probing to validate the detected differentiations. We also provide insight into when differentiations occur and how significant they are in the large ISPs we studied. Finally, we demonstrate that content and routing based differentiation can be easily implemented on today's commercial routers.

5.6.1 Content-based differentiation

Table 5.3 presents the detection results of content-based differentiation. We only listed the 4 ISPs that exhibits large degree of differentiation. We use the performance of HTTP as a baseline in comparison with the performance of each of the 4 remaining applications. For a particular application, the numbers in the "Paths" column are the number of ISP internal IP-level paths on which differentiation of the application is detected.

Surprisingly, these 4 large ISPs show clear evidence of differentiation of applications such as BitTorrent, PPLive, SMTP, and VoIP in Table 5.3. For instance, BitTorrent probes experience higher loss rate on 3794 (19%) paths in Tiscali. This is also true for SMTP probes on 573 (11%) paths in UUNet. In contrast, Sprint and UUNet treat VoIP probes preferentially on 172 (3.2%) and 203 (2.1%) internal paths. While content-based differentiation is known to exist in broadband ISPs, we are the first to detect such differentiation in backbone ISPs.

The percentage of internal paths with detected differentiation is a bit small for VoIP probes in Sprint and UUNet. This is likely due to two reasons: 1) the differentiation policy has limited deployment. In fact, we only detect differentiation of VoIP at one PoP in Sprint and UUNet respectively; 2) the effects of differentiation are evident only during certain periods, e.g.when links are congested. Thus, we may not always observe differentiation on certain paths even if they are configured with differentiation polices. We will study the correlation between detected differentiation and network load in §5.6.5.

Since we use 95% confidence interval in K-S test to detect differentiation, we want to understand whether this threshold is robust to inherent noise in the measurements. For this purpose, we randomly divide the loss rate samples of the same application measured on the same path into two equally-sized subsets. Then we apply K-S test on the two subsets and report the results in the "Same" column in Table 5.3. As expected, the number of paths that pass the test is significantly smaller than the corresponding number in the "Paths" column. This indicates the 95% confidence

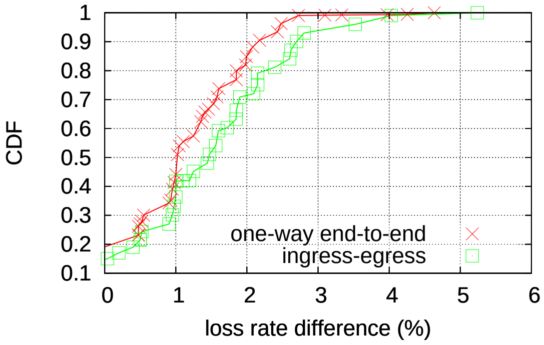


Figure 5.8: Validation using two-ended controlled measurement

interval is large enough to withstand noise in the measurements. In other words, the loss rate differences on the paths in the "Paths" column are more likely to be caused by traffic differentiations instead of measurement noise.

When we run K-S test on an application pair, each set in the pair contains roughly 1,000 loss rate samples. We want to study whether the set size is large enough to produce stable test results. In order to do this, we randomly select $\frac{1}{3}$ of samples from each set and run K-S test on the two subsets. The results in the "$\frac{1}{3}$" column in Table 5.3 are indeed very close to those in the "Paths" column, suggesting that 1,000 samples are sufficient for differentiation detection.

The remaining columns in Table 5.3 are used to further validate the results, which will be explained in §5.6.4.

5.6.2 Validation with two-ended controlled probing

As mentioned in §5.5, loss rate measured by TTL-based probing could be affected by various types of noise. We perform two-ended controlled probing to validate the content-based differentiation results presented in the previous section. Given all the

ISP	Previous-hop				Next-hop				Path pairs
name	AS pairs	δ_{tos}	TOS$_\delta$	$\frac{1}{3}$	AS pairs	δ_{tos}	TOS$_\delta$	$\frac{1}{3}$	in same ASes
Qwest	480, 11	85	25	449, 10	97, 1.6	84	21	-	6,0.1
UUNet	440, 2.4	48	94	471, 2.5	130, 0.7	100	88	-	90, 0.5
Sprint	1086, 21	89	86	1209, 23	3159, 15	92	84	3012, 14	11, 0.05
Verio	158,6	21	65	122,5	0	-	-	-	0
Level3	559, 16	98	79	516, 14	164, 4.9	97	70	129, 3.8	10, 0.3
Savvis	670, 10	71	41	693,10	103, 1.3	96	32	-	33, 0.4
AT&T	501, 9	77	81	574,10	109, 1.5	100	80	-	5,0.07
British	662, 17	99	80	608,15.6	93, 2.3	96	82	-	39, 1
Tglobe	1511, 30	67	90	1209, 24	102, 2	100	94	-	5,0.1
Above	51, 9	94	91	50,9	0	-	-	-	0

Table 5.4: K-S test results for routing-based differentiation.

PlanetLab node pairs, we first select a subset of them that traverse the ISP internal paths with detected differentiation. In total, we found 13 such pairs, all traversing the internal paths of Tiscali with differentiation against BT. Between each pair of nodes, we simultaneously measure the one-way end-to-end loss rate as well as the loss rate between ingress and egress of Tiscali with TTL-based probing, using both HTTP and BT probes. In Figure 5.8, the two curves labeled “one-way end-to-end” and “ingress-egress” correspond to the CDF of loss rate differences between HTTP and BT measured by two-ended controlled probing and TTL-based probing respectively. Clearly, the two curves match quite well, implying that the differentiation between HTTP and BT can also be confirmed by one-way loss rate measurements.

5.6.3 Routing-based differentiation

Table 5.4 summarizes our findings for the 10 ISPs which appear to carry out routing-based differentiation. For previous-hop AS based differentiation, the numbers in the “AS pairs” column are the number of previous-hop AS pairs between which differentiation is detected. Clearly, previous-hop AS based differentiation is commonly used by many ISPs, reflecting the fact that ISPs usually maintain different business contracts with their customers and peers. The number of previous-hop AS pairs exhibiting differentiation can be as large as 1511 (30%) in Teleglobe and 1086 (21%) in Sprint. In contrast, next-hop AS based differentiation is far less prevalent. Except for Sprint, all the other ISPs in the table show fewer cases of next-hop AS based differentiation. This is likely due to the clear advantage of previous-hop AS based approach in ease of implementation, e.g.an ingress router can simply mark packets based on their incoming interfaces.

To further confirm that the 95% confidence interval used for differentiation detection is robust to measurement noise, we apply K-S test on path pairs that traverse the same (AS_p, P_i, P_e, AS_n) tuple. These path pairs are not subject to previous-hop or next-hop AS based differentiation and should not pass the test. The last column in Table 5.4 shows only a small number of such path pairs pass the K-S test. This implies the loss rate differences between the AS pairs in columns 2 and 6 are most likely to be caused by routing-based differentiations rather than by measurement noise.

Simiar to the previous section, we run K-S test on each AS pair by randomly selecting $\frac{1}{3}$ of the samples in each set. The results in the "$\frac{1}{3}$" columns match those in the "AS pairs"columns quite well. This again confirms 1,000 samples are enough to produce stable test results. The remaining columns in the table are used for validation purpose, which will be covered in the next section.

The neighbors of an ISP can generally be classified into customers and peers based on whether the ISP receives payments from them. ISPs should have incentives to give customer traffic high priority. To confirm this conjecture, We employ the commonly-used relationship inference results by Gao [56] to classify the previous-hop ASes into customers and peers.

Among all the previous-hop AS pairs consisting of one customer and one peer, Table 5.5 shows the number of cases where customer's traffic receives better or worse treatment in columns 2 and 3 respectively. 7 ISPs either consistently or mostly give customer's traffic higher priority, confirming our conjecture. The remaining 3 ISPs (Sprint, British Telecom, and Teleglobe) appear to do the opposite. This could result from some special business agreements between these 3 ISPs and their peers.

We also investigate whether an ISP provides differentiated services to its customers. Columns 4 and 5 in Table 5.5 shows the number of customers who are assigned higher or lower priorities than the majority of the regular customers. Most ISPs have a small number of customers whose traffic experiences better performance, likely due to certain premium service they purchased from their ISPs. We perform similar analysis on the peers of each ISP and the results are in columns 6 and 7. Compared with customers, the results of peers are more mixed. Certain ISPs, e.g.UUNet, may even assign low priority to some of their peers. This could reflect the fact that they are dissatisfied with their existing agreements with those peers. Customers and peers of an ISP can use our results to tell whether their actual service quality matches their own expectations.

ISP	customer-peer pairs		customers		peers	
	+	-	+	-	+	-
Qwest	58, 10	7,1.2	7,4	0	3,1.7	5,2.9
UUNet	406, 2.6	0	0	0	0	12,15
Sprint	362, 12	541, 19	13,2.5	0	10,1.4	0
Verio	36,4	22,2.4	7, 3.3	0	4, 2.8	0
Level3	98, 13	13,1.7	10,2.1	2,0.4	0	4,2
Savvis	569, 15	0	0	0	4,10	0
AT&T	365, 12	0	0	0	8, 6	0
British	99, 5	232, 12	3, 4	0	3,3.5	2,5
Tglobe	134, 12	243, 23	5, 3.5	0	3, 5	0
Above	15, 10	0	1, 0.4	0	0	0

Table 5.5: Customer vs. peer in previous-hop AS based differentiation.

IP	DNS name	TOS	
		BT	HTTP
2 192.80.43.49	tuco.telcom.arizona.edu	0	0
3 192.80.43.65	morgan.telcom.arizona.edu	0	0
4 216.64.190.5	static.twtelecom.net	0	0
5 66.192.251.27	-	0	0
6 *213.200.80.94*	so-1-0-0.was11.ip.tiscali.net	128	0
7 213.200.80.26	so-3-0-0.lon12.ip.tiscali.net	128	0
8 89.149.186.185	xe-2-0-0.lon10.ip.tiscali.net	128	0
9 89.149.187.121	xe-0-0-0.bru20.ip.tiscali.net	128	0

Table 5.6: An example of content-based differentiation confirmed using TOS (from planetlab1.arizona-gigapop.net to 193.58.13.1)

ISP	Port
Tiscali	1214 (Napster), 4004 (PPLive), 4662 (eDonkey), 6881-6889 (BitTorrent), 6946, 6961-6969, 6999
Sprint	10, 5060 (VoIP)
Verio	179 (BGP), 16384 (VoIP), 25 (SMTP), 2525 (mail)
UUnet	25 (SMTP), 53 (DNS), 109 (POP3), 443 (IMAP), 1575, 5060 (VoIP)

Table 5.7: Applications ports used for TOS marking.

5.6.4 Correlation with TOS value

As previously illustrated in §5.2, traffic differentiation can be implemented in the router by marking the Type of Service (TOS) field in the IP header. We develop a method to reveal the TOS field marked by the routers along a path. We then study whether the observed traffic differentiation can be explained by different TOS values. Note that our probe packets trigger ICMP time exceeded messages from routers. These ICMP messages contain the IP header of the original probe packets, including the TOS values set by the routers. This allows us to correlate the loss rate differences with TOS value differences for each application pair and AS pair.

We start with an example of the TOS marking behavior of content-based differentiation. Table 5.6 illustrates the traceroute output from a PlanetLab node in University of Arizona to an IP address in AS3304 traversing Tiscali. The "TOS" column shows the TOS value of original probe packets extracted from ICMP replies. It is clear that the TOS value of BT probes were set to 128 by the router at the sixth hop while that of HTTP probes are always 0. We further conduct controlled experiments to infer which packet fields are used to perform TOS marking. We vary the composition of probe packets by chaning destination ports or faking application payloads. In this example, the marking is done purely based on destination port number, e.g.packets with the default BT port and fake payloads are still marked.

In following analysis, we assume an ISP has a consistent policy of associating a TOS value with a fixed priority. However, we do not assume that a large TOS value is always associated with a high priority. We first need to infer the relationship between TOS values and priorities. Starting with all the pairs that pass K-S test, we compile a list of all the distinct TOS values observed in a target ISP. We then construct a mapping from TOS values to priorities in a way that the loss rate differences between the pairs with differentiation can be best explained. More specifically, given a pair with differentiation, if the TOS value of the first set has higher priority than that of the second set, the former should have lower loss rates as well.

Once a mapping is constructed for each ISP, we correlate TOS value differences with detected differentiation in two ways. First, we compute δ_{tos}, which is the percentage of pairs with detected differentiation that can be explained by differences in priorities and TOS values. The results are in the "δ_{tos}" columns in Tables 5.3 and 5.4, where "-" means no TOS marking is used. Clearly, there is a strong correlation between detected differentiation and priority differences inferred from TOS values. δ_{tos} is 100% for pairs with content-based differentiation (Table 5.3). For pairs with previous-hop AS based differentiation, δ_{tos} is over 80% in 5 ISPs (Table 5.4). Note that δ_{tos} is below 100% in many target ISPs. This could be caused by resource constraint at certain links. For instance, traffic traversing an under-provisioned link may persistently experience high loss rates, even though the target ISP does not "intentionally" treat it with low priority.

Second, we compute the percentage of pairs with different priorities that are detected to have traffic differentiation. The results are in the "TOS$_\delta$" columns in Tables 5.3 and 5.4. Overall, TOS$_\delta$ is pretty high. Only VoIP in UUNet has a TOS$_\delta$ smaller than 80% in Table 5.3. For previous-hop AS based differentiation, TOS$_\delta$ exceeds 80% in 6 target ISPs in Table 5.4. The reason that TOS$_\delta$ is not 100% is likely due to the fact that the effects of differentiation are perceptible only under certain conditions, e.g. when there is resource competition. As a result, we may miss certain pairs which are indeed configured with differentiation policies. We will study the relationship between detected differentiation and network load in §5.6.5.

Application	ISP	High loss %	Low loss %
BT	Tiscali	96	45
PPLive	Tiscali	100	62
VOIP	UUNet	106	65
VOIP	Sprint	100	51
SMTP	UUNet	100	53
SMTP	Verio	100	14

Table 5.8: Network load effects for content-based differentiation: % of pairs with detected differentiation compared with using all samples.

For the 4 ISPs verified to use TOS markings for content-based differentiation, we further analyze which packet fields are used to perform TOS marking. Such information is especially useful for customers who want to circumvent ISPs' differentiation policy. We conduct controlled experiments by changing packet headers and application payloads in probe packets. *Surprisingly, we found all the 4 ISPs simply use destination port to mark TOS value.* By enumerating different destination ports, we can clearly observe changes in TOS markings. Table 5.7 lists all the destination ports which are used by the 4 ISPs for TOS marking. For instance, besides SMTP, UUNet also treats traffic destined to other email ports differently, e.g.POP3 and IMAP. We plan to perform a more comprehensive study on whether ISPs use rules other than destination port to construct their differentiation policy as future work.

5.6.5 Correlation with network load

Given the strong evidence of traffic differentiation performed by some large ISPs using packet contents (application types) and routing (previous-hop AS) information, we now investigate whether there exists other factors that may affect traffic differentiation. In particular, if ISPs intend to use differentiation to conserve limited resource

ISP	High loss %	Low loss %
Qwest	91	24
UUNet	100	70
Sprint	102	45
Verio	100	20
Level3	100	74
Savvis	96	16
AT&T	100	23
British	100	47
Tglobe	86	65
Above	94	39

Table 5.9: Network load effects for previous-hop based differentiation: % of pairs with detected differentiation compared with using all samples.

in their networks, we should be able to observe a strong correlation between network load and traffic differentiation. For instance, an ISP may throttle BT traffic only when its bandwidth exceeds 100Mbps.

Although we cannot measure network load directly, we can observe its effects in terms of loss rate. For each application or AS pair, we sort the samples in each set based on loss rate value and partition the samples into two equally-sized groups: high-loss *vs.* low-loss. We then perform K-S test both between the two high-loss groups and between the two low-loss groups. Tables 5.8 and 5.9 summarize the test results. For ease of comparison, we show the relative number of pairs passing the K-S tests compared to the corresponding numbers in Tables 5.3 and 5.4 where we do not distinguish between high-loss and low-loss groups. The relative numbers in the high-loss group are significantly higher than those in the low-loss group, clearly supporting our conjecture that ISPs perform load-sensitive traffic differentiation. This also highlights the importance of observing ISPs' behavior continually for detecting traffic differentiation.

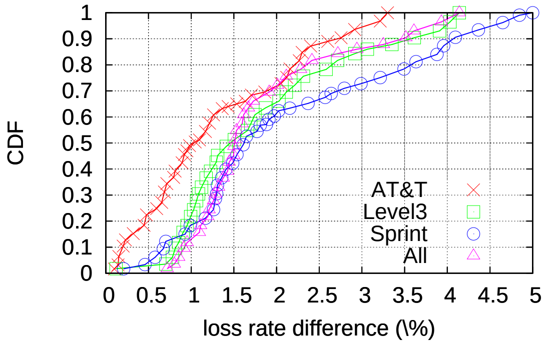


Figure 5.9: Loss rate difference for previous-hop AS based differentiation.

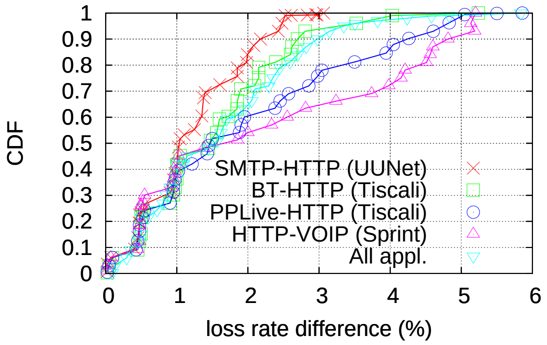


Figure 5.10: Loss rate difference for content-based differentiation.

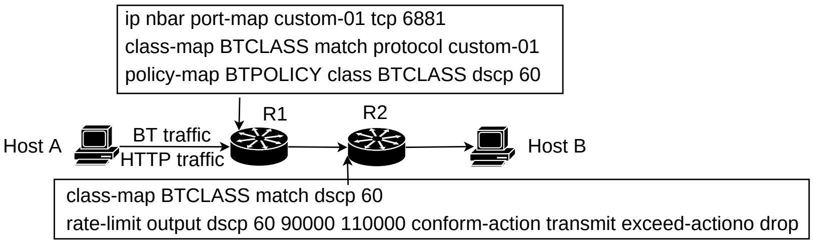
5.6.6 Degree of differentiation

The statistical tests we devise can systematically detect whether there exists differences between two loss rate distributions. We now study whether the actual loss rate differences are significant enough to affect the perceived performance of TCP-based applications. For each AS pair with previous-hop AS based differentiation, we first compute the mean loss rate of each set. We then compute the absolute loss rate difference between the two mean loss rates. Figure 5.9 plots the CDF of absolute loss rate differences of all the AS pairs in three target ISPs. Among them, AT&T has the smallest loss rate differences, mostly under 3%. In contrast, the differences are much more evident for Sprint. Nearly 10% AS pairs have loss rate difference over 4%. Such large loss rate difference will certainly lead to perceptible performance difference for many TCP-based applications.

Figure 5.10 illustrates the CDF of absolute loss rate differences of the application pairs included in Table 5.3. For each application pair, the absolute loss rate difference is computed as the difference between the mean loss rate of an application (e.g.BT) and that of HTTP. Clearly, the degree of content-based differentiation varies significantly across different applications and ISPs. For instance, UUNet treats SMTP only slightly worse than HTTP. Their loss rate differences are smaller than 2% on nearly 90% of paths. In comparison, Sprint gives VoIP much higher priority than HTTP, possibly reflecting their desire to meet the QoS requirements of the VoIP service provided by themselves. Another interesting observation is that Tiscali appears to differentiate multiple classes of applications. Although both BT and PPLive experiences worse performance than HTTP, the loss rates of PPLive are even much higher than those of BT. Since PPLive is a real-time video streaming application, its users are much more susceptible to high loss rates than BT users who normally download files in the background.

5.6.7 Implementation of differentiation in router testbed

In this section, we demonstrate the feasibility of implementing and enforcing traffic differentiation in today's commercial routers. As shown in Figure 5.11, we set up our own experimental testbed using two high-end routers (Cisco 7300 and 12000) running the latest IOS 12.3 from the Schooner testbed [11]. Host *A* transmits BT and HTTP



```
ip nbar port-map custom-01 tcp 6881
class-map BTCLASS match protocol custom-01
policy-map BTPOLICY class BTCLASS dscp 60
```

```
class-map BTCLASS match dscp 60
rate-limit output dscp 60 90000 110000 conform-action transmit exceed-actiono drop
```

Figure 5.11: Router testbed setup

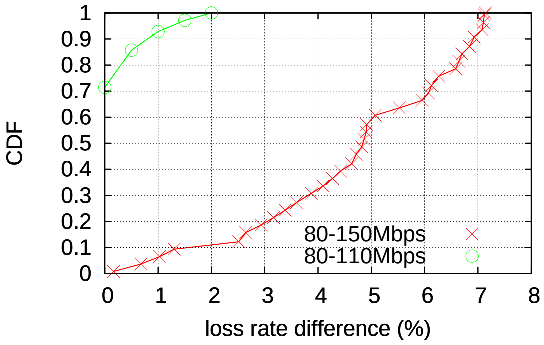


Figure 5.12: Loss rate difference on router testbed

traffic to host B via R_1 and R_2. All the machines and routers are connected using Gigabit Ethernet links. To configure the routers for port-based differentiation, we define a port-map on R_1 to capture all the packets with the default BT port and mark their TOS field using policy map. Interestingly, we found the default router configurations already include pre-defined port-maps for applications such as Napster, Kazaa, SMTP, *etc.* [36], which greatly simplifies the work of configuring differentiation for these applications. The actual router commands used in the Cisco command line interface (CLI) are shown in Figure 5.11. Similarly, to implement previous-hop AS based differentiation, we can easily mark packets based on incoming interfaces by changing the definition of *class-map* to *class-map NEIGHBOR match interface GigabitEthernet 1/0*. We configure R_2 to prioritize traffic on its incoming interface using weighted random early drop (WRED) queuing.

In § 5.6.5, we observed that the effects of traffic differentiation are more perceptible when network load is high. To illustrate this, we measure the variations of loss rate differences between HTTP and BT as we control the sending rate on A. The configurations of R_1 and R_2 remain the same throughout the experiments. R_2 will restrict the BT bandwidth to be within 110Mbps. Figure 5.12 shows the absolute loss rate differences between BT and HTTP under two different ranges of sending rates. When the sending rate is high (80 - 150Mbps), the loss rate differences can go up to 7%. In contrast, when the sending rate is below the bandwidth limit (80 - 110 Mbps), the loss rate differences become negligibly small. We also measure the overhead induced by the differentiation configurations on R_1 and R_2. From the SNMP logs, we observed little changes in the CPU utilization on R_1 and R_2 when we disable or enable the differentiation configurations. This indicates the overhead of enforcing either types of differentiation is small.

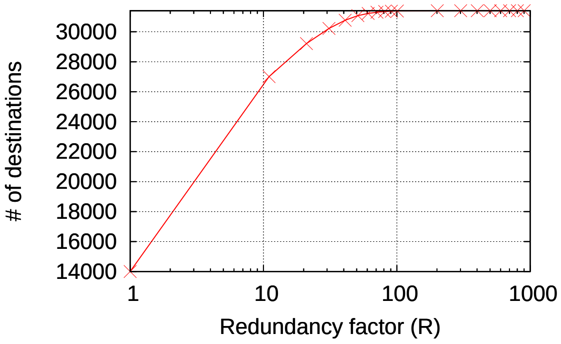


Figure 5.13: Impact of redundancy factor.

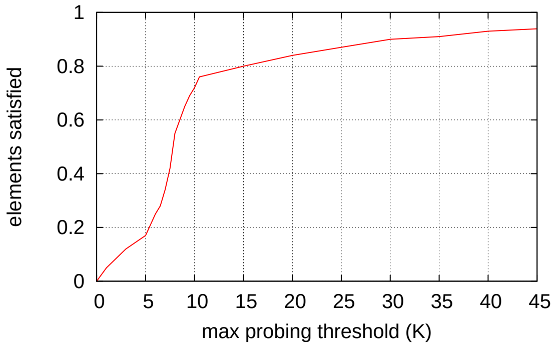


Figure 5.14: Impact of the maximum probing threshold.

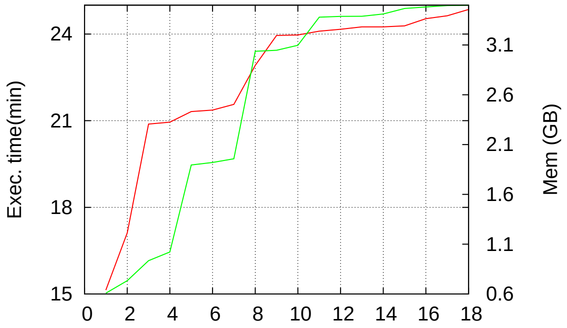


Figure 5.15: Execution time and memory usage of path selector.

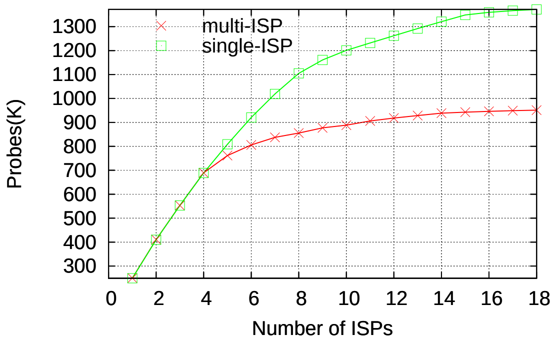


Figure 5.16: Probing overhead under single-ISP *vs.* multi-ISP path selection.

5.7 System evaluation

In this section, we study the parameter settings and system performance in NVLens. We will explain the choice of redundancy factor and maximum probing threshold (defined in §5.3). We will also evaluate the resource usage of NVLens in terms of network, memory, and CPU. Our evaluation results demonstrate the feasibility of deploying NVLens as a lightweight tool for continually detecting traffic differentiation in multiple large ISPs simultaneously.

Parameter settings The path selection process of NVLens is controlled by two predefined parameters: the redundancy factor R and the maximum probing threshold m. R determines the number of distinct paths that will traverse each element. An element can be a three-tuple of $(src, ingress, egress)$ or $(ingress, egress, dst)$. Figure 5.13 shows the maximum number of destinations assigned to a prober increases with R and remains the same once R exceeds 100. This means when $R \geq 100$, the redundancy of each element is no longer restricted by R but by the set of destinations the probers can probe. We set $R = 100$ to obtain the best coverage.

NVLens imposes a maximum probing threshold m to prevent a prober from being assigned too many probing destinations. This may cause the actual redundancy of certain elements to be smaller than R. Figure 5.14 shows the fraction of elements whose actual redundancy reaches $R = 100$ under different m. The fraction number grows slowly when m exceeds 10K. We choose $m = 10K$ to attain a reasonable balance between element redundancy and prober overhead. Note that the redundancy of certain elements can never reach R because the number of distinct paths traversing an element is inherently limited by the set of source-destination pairs covered by NVLens.

Performance evaluation In NVLens, the number of destinations probed by each prober ranges from 6K to 10K. This corresponds to a bandwidth usage from 17Kbps to 443Kbps per prober. The multi-ISP path selection consumes most of the execution time and memory compared to other components in NVLens. Since a path can only traverse a limited number of elements, the time and space complexity of the path selection is $O(p^2)$ and $O(ep)$. Here, p is the number of source-destination pairs and e is the number of elements.

We evaluate NVLens on a commodity server with eight 3.0GHz Xeon processors and 8 GB memory running Linux 2.6.18 SMP. Figure 5.15 illustrates the execution time and memory usage of NVLens as the number of ISPs increases. At 18 ISPs, it takes 3.5GB memory for NVLens to store 182M elements. The execution time of each run of path selection is around 25 minutes, which is only 20% of a probing interval. This means the path selection process can keep up with measurement speed. To demonstrate the benefit of multi-ISP path selection, Figure 5.16 compares the total number of paths probed under single-ISP path selection *vs.* multi-ISP path selection. The latter reduces the probing overhead by almost a third when 18 ISPs are being measured.

5.8 Summary

Broadband ISPs and wireless carriers are known to rate-limit or block bandwidth-intensive applications such as P2P file sharing and video streaming. Such traffic differentiation may severely degrade the application performance experienced by users.

In this chapter, we presented the NVLens system to detect content- and routing-based traffic differentiation in backbone ISPs by taking loss measurement from end hosts. NVLens employs an intelligent probing scheme to attain rich coverage of ISP internal paths while maintaining reasonable measurement overhead. It identifies significant performance gap between different types of traffic using statistical hypothesis tests.

We deployed NVLens on PlanetLab to study 18 large ISPs across 3 continents over 10 weeks. We find 4 ISPs perform differentiation on 4 distinct applications and 10 ISPs perform previous-hop AS based differentiation, evidenced by up to 5% absolute loss rate differences. The degree of differentiation increases with network load. Some ISPs appear to carry out application-based differentiation simply based on port numbers irrespective of packet content. The loss rate differences are often associated with different TOS values in the traffic marked by ISP routers. Our work serves as an important step towards increasing the transparency of the Internet.

So far, I have demonstrated the end users' capability to accurately diagnose problems caused by both routing changes and ISP policy changes. Ultimately, the monitoring and diagnosis results should be used for mitigating the damage or preventing being affected in the future. In the next two chapters, I will present the two applications to mitigate effectively in both short-term and long-term periods.

Based on the observed traffic differentiation, NVLens can further infer the policies used by the ISPs to implement differentiation as well as the location of its enforcement. The methodology is general and can be easily extended to discover other types of differentiation, e.g.IPSec vs. non-IPSec. Such information is essential for end-systems to make more informed decisions for selecting routes and ISPs, applying encryption or routing through proxies to bypass unwanted differentiation.

Even if ISPs are aware of techniques used by NVLens to detect potential traffic differentiation in backbones, they cannot easily evade our detection. The probe packets are constructed using real traffic traces and are difficult to distinguish from actual data traffic. Unless ISPs perform stateful TCP flow analysis, it is challenging to identify and preferentially treat our probe traffic. In the future, we plan use two-ended controlled experiments to mimic actual TCP flows.

Revisiting the literature review in Chapter II, the work in this chapter falls into end-host based diagnosis category. We examine one type of performance degradation, i.e.ISP policy-induced disruptions, which hasn't been examined in the past. Combining with the system in Chapter IV, we demonstrate the ability for end-host to diagnose two types of disruptions, i.e.routing-induced and policy-induced disruptions. The consequence of traffic differentiation can be long delay or packet loss in the data plane. So far, we have been using purely active measurement approach. The active approach has its fundamental limitations in terms of non-trivial overhead. To reduce overcome this limitation, in Chapter VI we propose a new methodology that combines the active approach together with passive approach to achieve more efficient measurement.

CHAPTER VI

Measuring and Predicting the Impact of Routing Changes

6.1 Overview

Internet routing dynamics directly influence the data plane, i.e.the packet forwarding behavior. Previous measurement studies [73, 85, 113, 123] have already shown that routing changes can cause transient disruption to the data plane in the form of packet loss, increased delay, and forwarding loops. In this work, we enhance our understanding of the impact of routing dynamics on the data plane performance in two dimensions. First, we develop an efficient framework enabling a more comprehensive study of routing changes that are not limited to just specific prefixes as in previous studies. Second, we identify the predictability of observed performance degradation in relation to the properties of routing updates and subsequently develop a model to accurately predict the performance impact of future updates.

We use the term *data plane failures* to describe severe performance degradation on packet forwarding manifested as reachability loss or forwarding loops. Our study focuses on data plane failures primarily caused by routing changes, as understanding the impact of routing dynamics on data plane performance is critical to the deployment of real-time applications such as Voice over IP (VoIP) and moreover provides insights into improved network operations.

Routing changes on the Internet are mostly caused by failures or configuration changes. They occur quite frequently. At the interdomain level, one can easily observe more than 10 updates per second to a wide range of destinations from a large tier-1 ISP such as Sprint using publicly available BGP data from RouteViews [13]. Motivated by such active routing dynamics on the current Internet, our study develops a methodology to identify properties of updates that cause data plane failures and characterize the location, duration, and stability of these failures.

Data plane failures are often caused by inconsistent forwarding information of routers involved in routing changes [123]. During routing convergence, some routers may lose their routes [122] or have invalid routes [73]. Routing policies, timer configurations, and network topologies are just some of the contributing factors [122, 123]. For instance, transient loops can be caused by temporarily inconsistent views among routers. Persistent loops are more likely due to misconfigurations [127]. We do not attempt to identify the cause of observed failures due to lack of information but instead search for patterns to help predict the impact of routing changes on data plane performance. Such a prediction model can improve route selection.

To achieve a comprehensive characterization of many diverse routing changes, we develop an efficient and novel measurement framework deployed at each vantage point with access to real-time BGP routing updates. Light-weight probing is triggered by locally observed routing updates. The probing target is an identified live IP address within the prefix associated with the routing change. Compared to modeling or simulation based approaches [26, 99, 142] to understand the impact routing dynamics on data plane performance, our measurement-based approach does not make simplifying assumptions and provide empirical evidence of such impact.

Given that probing is triggered directly by routing updates, it may be counter-intuitive why the observed data plane performance may still be impacted by the seemingly converged route. In some cases, the routing change is still ongoing, often manifested by subsequent updates to the same destination prefix. Given the scale of the Internet, some routing changes may impact many routers and cause delayed convergence [73]. Thus, even if locally the route to a destination appears to be converged to a stable route, data plane performance may still be seriously affected. This is supported by previous work showing that BGP messages sometimes preceded observed path failures in the order of minutes [52].

We deployed our measurement framework at six geographically distinct locations with different upstream providers for a period of 11 weeks. Using our collected set of 604,925 live IPs which belong to 48% of prefixes and 53% of ASes, we analyzed 47%-55% of all observed updates corresponding to 46%-51% of observed prefixes in routing updates across different vantage points.

We summarize our main findings by including a range of results to represent all six vantage points studied.

- Many prefixes became unreachable shortly after respective routing changes. They account for 39%-45% of probed updates, covering 72%-86% of probed prefixes. These prefixes belong to 35%-42% of all announced prefixes, originating from 39%-42% of all ASes. Stub ASes are more likely impacted. Unreachable incidences are usually transient: 84%-91% of them lasting less than 300 seconds. The failure location occurs roughly equally likely along the path.

- Among the unreachable incidences, a non-negligible fraction exhibits forwarding loops. This contributes to 4%-8% of probed updates, covering 36%-51% of probed prefixes. These loops impact 17%-24% of all announced prefixes, originating from 27%-34% of all ASes. Most loops are short-lived: 60% of them lasting less than 300 seconds. Loops are more likely to appear within large ISPs.

- Given a prefix and its identified responsible AS where traceroute stops or loop occurs, we identify over 51%-54% of probed updates to be predictable for causing reachability loss, and 49%-58% for causing loops. For such prefixes, our prediction model achieves a prediction accuracy of 90% with a false positive rate of 15% for unreachable incidences and a prediction accuracy of 80% with a false positive rate of 12% for loops. In general, prefixes originating from stub ASes and smaller ISPs are more predictable; responsible ASes for such predictable prefixes also tend to be near the edge of the Internet.

Aside from measurement findings, our main contribution is a framework to efficiently measure the impact of routing dynamics on data plane performance. Based on identified inherent stability of routing changes, we develop a methodology to predict impact of future routing updates. The ability to accurately predict routing-induced data plane failures is directly useful for applications such as overlay route selection and backup path selection.

This chapter is organized as follows. Section 6.2 introduces our measurement methodology. Experiment setup is described in Section 6.3. We provide detailed data analysis on probing results in Section 6.4. In Section 6.5, we present a prediction model.

6.2 Measurement Methodology

We describe our measurement methodology to enable efficient characterization of the impact of locally observed BGP routing updates on the data plane performance from the local network to the relevant destination networks.

6.2.1 Terminology

We first introduce our terminology. We use the term *data plane* to refer to the packet forwarding behavior on the Internet. *data plane failures* describe severe data plane performance degradation in the form of reachability loss or forwarding loops. The *control plane* computes the routing state of network elements performing packet forwarding. On today's Internet, inter-domain routing involves distributed router computation within routers of different networks.

To describe probing results, we use the term *probing incidence* to mean a set of probes to the relevant destination prefix triggered by a BGP update of the prefix. Three ping requests optionally followed by a traceroute probe are sent for each prefix probed. The destination is deemed *reachable* if any ping reply returns or the traceroute response contains interface IPs belonging to the prefix. It is *unreachable* otherwise.

6.2.2 Data Collection

There are two required data sources: control-plane BGP updates and data plane active probes. For each monitored location, local real-time BGP data are analyzed to identify probing destinations. BGP data can be obtained by setting up a monitoring BGP session using software such as Zebra [5] with a BGP router with a default-free routing table in the local network. To differentiate between unreachable destinations and blocked probes due to firewalls, we must identify at least one *live IP* that responds to ping or traceroute requests for each prefix probed. Besides active probing [131], such data can be gathered passively from various server logs, e.g.Web and DNS server logs, or traffic traces.

6.2.3 Active Probing Methodology

Figure 6.1 depicts the probing architecture for one vantage point consisting of a BGP analysis host identifying probe targets based on the local BGP feed and a probe host in the same network for performing probing triggered by routing updates. The list of live IPs is continuously updated. To identify persistent failures and verify live

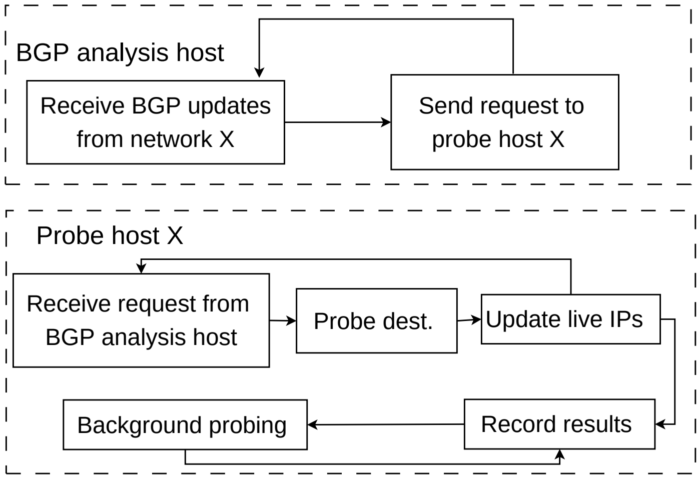

BGP analysis host
Receive BGP updates from network X
Send request to probe host X
Probe host X
Receive request from BGP analysis host
Probe dest.
Update live IPs
Background probing
Record results

Figure 6.1: Active probing architecture for vantage point X (both functionalities can be implemented on the same host).

IPs' responses, background probing is done.

6.2.3.1 Probing Methodology

Unlike previous studies, our probing is designed to be light-weight to scale to many destinations covering most observed updates. Therefore, we focus on coarse-grained performance metrics associated with reachability. We are nevertheless limited to probing only prefixes for which we have identified a live IP. We plan to remedy this in the future.

We describe the detailed probing steps. Triggered by a routing update, three ICMP-based ping requests are first sent to the corresponding live IP. We randomly choose the IPs belonging to the given prefix and regularly update IP liveness. Three is chosen to balance the overhead and packet loss probability. If any ping reply returns, the destination is considered *reachable*. Otherwise, traceroute is performed. If the traceroute response contains an IP belonging to the probe destination prefix, the

Category	Tier-1	Tier-2	Tier-3	Tier-4	Tier-5
Num of ASes (Pctg relative to all ASes in each tier)	20 (90%)	173 (80%)	1092 (78%)	1235 (80%)	7136 (52%)
Num of prefixes	3045	4672	10034	9424	16727
Num of IPs	73670	119136	134982	126818	116643

Table 6.1: Diversity of networks covered by our collected live IPs.

destination is deemed reachable. Otherwise, ping and traceroute probes are continuously sent after each other as soon as the previous probe finishes, until the destination or a timing limit is reached as described later.

6.2.3.2 Probing Control

Given the potential high frequency of routing updates, we take measures to avoid overloading the probe host and the destination networks probed. The resources under consideration are CPU and memory resources of the probe host, and network bandwidth of both the probe host and targets. Multiple probe hosts can be used. We make explicit trade-offs between probing coverage and consumed resources.

The first measure is to ignore routing updates caused by the BGP session reset of the monitoring session using known techniques such as [132], as such updates do not reflect true routing changes. As a second measure, we impose a limit on the *maximum probing duration* for each destination prefix. Probing is performed as long as the target is deemed unreachable until this limit is reached. Moreover, at most one IP from each prefix is probed by a single host at any time.

Probe requests may not be serviced immediately due to unfinished probing. We impose a *maximum wait time* between the time an update is received and the time its probe request is initiated, as excessive delays prevent us from effectively capturing the impact of routing changes. As future work, we plan to explore other ways to reduce probing overhead, e.g.by probing based on unique AS paths.

6.3 Experiment setup

We describe the experiment setup based on our measurement methodology.

6.3.1 Data Collection

We set up a software router using Zebra [5] to serve as the BGP monitor to obtain live BGP feeds from six distinct locations with different upstream providers mostly in the U.S.: Michigan, Massachusetts, New York, Illinois, Washington, and Amsterdam. They belong to the PlanetLab [102] and the RON project [17]. Combining active probing [131], DNS logs, and five days of Netflow data from a Tier-1 ISP network, we collected 604,925 live IPs covering 48% of all announced prefixes and 53% of all ASes. Using the tier ranking defined in [115], where a lower tier means large ISPs and tier-5 refers to stub or customer ASes, we illustrate the diversity of collected IPs in Table 6.1. The set is shown to cover a large percentage of ASes in different tiers. The results presented span an 11-week period from May 3 until July 19, 2006.

6.3.2 Probing Control

We limit the maximum probing duration to be 300 seconds as most BGP routing changes converge within about three minutes based on previous studies [73, 85]. Our own measurements described later in Section 6.4 also show that about 90% of reach-

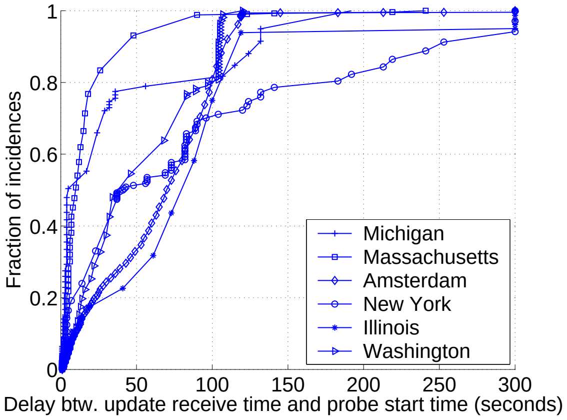


Figure 6.2: Probing delay distribution for each BGP feed: Most delays are within 100 seconds.

ability problems last less than 300 seconds. To ensure our characterization captures the effect of routing dynamics on the data plane, we limit the maximum wait time to be 300 seconds. Background probing is performed to ensure each live IP is probed at least once every 300 seconds.

6.3.3 Probing System Performance

During the 11 weeks of study, the average probing rate is only about 2 updates per second for each feed with a maximum rate of 11 updates per second. Probing duration varies from less than 10 seconds to the limit of 300 seconds.

Figure 6.2 plots the distribution of probing delays for each probing location. The delay is computed as the time difference between the probe time and the update receive time. The figure shows that at least 80% of updates are probed within 100 seconds for most feeds. For some locations, the delays are mostly between 50 to 100 seconds. Only 6% of updates are not probed due to the maximum wait time constraint.

To prevent aggressive probing, we measure the probing rate. We found that 80% of the difference between two consecutive probes for the same IP is larger than 300 seconds, with a minimum difference of around 100 seconds. This shows that our system did not overload the destination networks probed.

6.3.4 Probing System Limitations

We discuss the limitations of our probing methodology to understand the potential bias introduced in our results. First, the data presented later correspond to probing triggered by routing announcements only. We also probed after route withdrawals, as such prefixes can still be reachable due to covering prefixes: 1.4%-2.1% of withdrawn prefixes are reachable, while less than .013% of withdrawn prefixes are unreachable despite the presence of covering prefixes. But most of them recover within 300 seconds. Second, our probe delays are mostly within 100 seconds. Thus, we focus on serious data plane failures lasting for at least 100 seconds.

The third limitation is that we do not differentiate between performance degradation due to routing changes from other possibly unrelated causes such as congestion. Given that our probing immediately follows routing updates, the observed performance degradation could also coincide with other events. However, if a destination consistently experiences performance degradation following routing changes, such degradation may likely be caused by routing dynamics.

Although our probing uses simple ping and traceroute probes, we try to overcome limitations of measurement tools. For example, we distinguish unreachable cases caused by routers disabling ICMP replies from unreachable end hosts using history information.

6.4 Characterizing Data Plane Failures

During a routing event such as link failures or recoveries, packet forwarding is likely disrupted. This is likely caused by some routers temporarily losing their routes to the destination. Moreover, even without transient failures in the control plane, i.e.every router has a route to the destination, the route may not be valid due to routing inconsistency. Next, we characterize data plane transient failures using "reachability" as the performance metric. This is motivated by the fact that gain or loss reachability will cause the most severe impact on data plane performance.

6.4.1 Overall Statistics

We conducted Internet experiments over the period of 11 weeks from May 3, 2006 to July 19, 2006. Table 6.2 shows the overall statistics. We found that 42% of probing incidences are unreachable, affecting 73.5% of destination prefixes and 63% of destination ASes probed in our experiments. In addition, about 14% of the unreachable incidences are caused by loops, affecting 24% of destination prefixes and 34% of ASes probed.

		Incidence	Prefix	AS
Unreachable	Loop	185728 (6.0%)	21821 (23.9%)	5024 (33.5%)
	Other	1129014 (36.3%)	66321 (72.8%)	5802 (38.7%)
	All	1314742 (42.3%)	66883 (73.5%)	9559 (63.0%)
Reachable		1796392 (57.7%)	75578 (83.1%)	14870 (98.0%)

Table 6.2: General statistics over the period of 11 weeks

6.4.2 Reachability Failures

6.4.2.1 Destination Networks Impacted by Failures

We classify destination ASes experiencing reachability loss according to their tiers and geographic locations. Table 6.3 shows the top 10 destination ASes which encounter the most unreachable incidences. We observe that most of them are stub ASes, i.e.customer ASes. Moreover, we found that many unreachable incidences affect a small portion of destination prefixes and ASes observed in our routing updates. For example, as shown in Figure 6.3, 80% of unreachable incidences impact only 30% of prefixes and 10% of ASes, respectively.

Identifying the failure location along the path helps us understand whether the problem usually happens close to the destination networks. If failures occur near or within destination networks, multi-homing or overlay routing cannot bypass such failures. We approximate the location of a data plane failure as the IP hop where the traceroute probe stops.

Figure 6.4 shows that the normalized hop count is evenly distributed along both the IP level and AS level path. The hop distance is normalized by the hop count

ASN	Unreachable Incidences	Prefixes	AS Name	Tier	Primary Country
25543	112784 (8.6%)	34	FasoNet-AS ONATEL/ FasoNet's AS	5	Burkina Faso
4134	110787 (8.4%)	590	CHINANET-BACKBONE No.31 Jin-rong Street	2	China
19982	107709 (8.1%)	3	TOWERSTREAM-PROV Towerstream	4	United States
8866	45840 (3.4%)	72	BTC-AS Bulgarian Telecom Company	3	Bulgaria
9121	43021 (3.2%)	423	TTnet Autonomous System	3	TURKEY
8011	41768 (3.1%)	39	CoreComm - Voyager, Inc.	4	United States
22543	37267 (2.8%)	16	PIXELWEB Pixelweb	5	Canada
4595	36300 (2.7%)	8	ICNET ICNet/ Innovative Concepts	5	United States
17974	35573 (2.7%)	369	TELKOMNET-AS2-AP PT TELEKOMUNIKASI	5	Indonesia
4314	28951 (2.2%)	20	CommNet Data Systems, Inc.	3	United States

Table 6.3: Top 10 destination ASes experiencing most unreachable incidences.

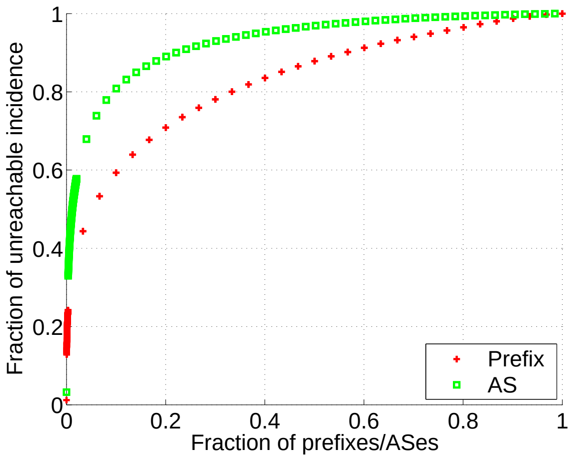


Figure 6.3: Destination prefixes and ASes affected by reachability problems.

of the reachable path before the incidence. Note that the last hop of the stopped traceroute may not be where the problem resides since absence of traceroute replies may be due to firewalls or routers disabling ICMP replies. We differentiate such cases by examining whether routers in a particular AS ever replied with ICMP packets in

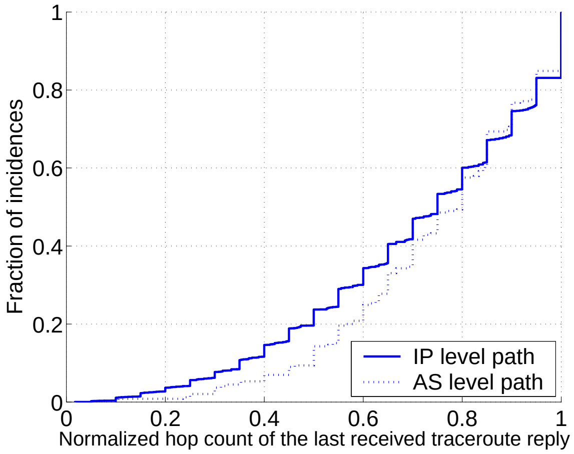


Figure 6.4: Normalized hop distance btw. the source and the last received traceroute reply.

history data. Such an AS is expected appear in the data path based on history or BGP data. Furthermore, we can usually assume that an AS applies a uniform policy regarding ICMP for all its routers [92].

6.4.2.2 Failure Duration

We compute the duration of reachability loss to be the period starting from the time when the update is received to the time that the destination is reachable by probing. Figure 6.5 shows the cumulative distribution of the duration of unreachable incidences. We found that most such incidences last than 300 seconds. They are likely due to transient routing failures [122] or routing convergence delays. However, 10% unreachable incidences last longer than the maximum probing limit of 300 seconds. They may be caused by other factors such as configuration errors and path failures. The observed reachability disruption lasting a few hundred seconds is expected to have serious performance impact on real-time applications such as Voice over IP.

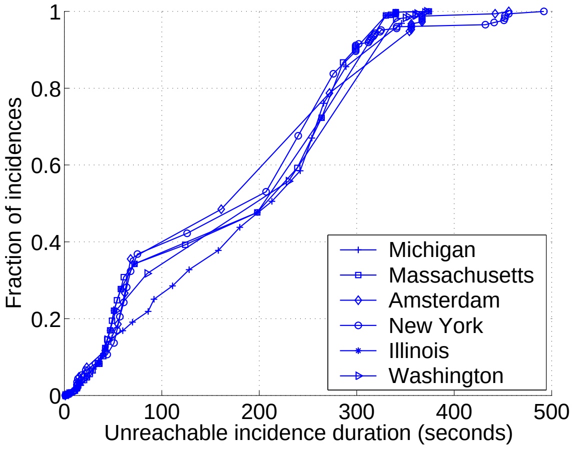


Figure 6.5: Duration of unreachable incidences.

6.4.2.3 Failure Predictability

Routing incidences and their corresponding impact on certain destination networks can be predictable. For a given destination prefix D, we define the *appearance probability* of D as the probability of an unreachable incidence occurring with any routing update to D. We define the *conditional probability* of D conditioned on an AS or an AS path segment as the probability of an unreachable incidence occurring under the condition of observing a routing update to D through a particular AS or an AS path segment. Moreover, we define the *responsible AS* for an unreachable incidence to be the AS where traceroute stops.

Figure 6.6 shows the CDF of the appearance probability and the conditional probability conditioned on the responsible AS. Around 30% of the prefixes have unreachable appearance probability of larger than 0.5. This indicates that the reachability loss is difficult to predict for most prefixes upon observing a routing update of that prefix. However, the corresponding plot for the conditional probability (conditioned

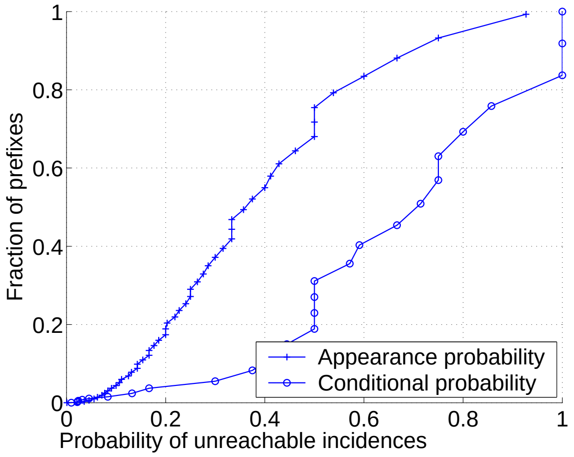


Figure 6.6: Appearance probability and conditional probability (conditioned on the responsible AS) of unreachable incidences.

on the responsible AS) is about 80%. This indicates that, given a routing update to a destination and the responsible AS, unreachable incidences can be much more predictable. By comparing the looping AS path with the normal AS path obtained from background probing, we can estimate the responsible AS's AS level hop count to the destination. 95.9% of these responsible ASes are at least one hop away from the destination. Therefore, taking alternate path might be possible to bypass the problem.

6.4.3 Forwarding Loops

We now focus on a subset of unreachable incidences – forwarding loops, which have been widely studied [60, 98, 100, 113]. It has been shown that transient loops can be caused by inconsistent or incomplete views among routers during routing convergence [143], while persistent loops are more likely a result of configuration errors [127]. In our experiments, we identify loops in traceroute and compare path from background probing with path from triggered probing to detect persistent loops and to exclude loops caused by measurement artifacts. We find only 0.027% forwarding loops are persistent. We focus on transient loops in the rest of this section.

6.4.3.1 Destination Networks Impacted by Loops

Figure 6.7 shows the fraction of destination prefixes and ASes impacted by forwarding loops. Similar to unreachable incidences, we observe that the distribution of loop incidences across destination prefixes and ASes are very skewed. For example, top 10% of prefixes and ASes observed in our routing updates experience 60% and 80% of forwarding loops, respectively.

For each loop incidence, we consider the ASes where the loop occurs as the *responsible ASes*. We observe that 98% of the loop incidences are intra-AS loops, i.e.the IPs involved in the loop are within one AS. Table 6.4 shows the top 10 responsible ASes for loop incidences. Interestingly, we observe that most of these ASes are tier-1 ASes. This is because large ASes in the core of the Internet have more complicated routing policies, potentially more complex routing dynamics, and larger network diameters translate to longer delays for propagating updates. All these factors can cause more transient failures within such networks [60, 122].

6.4.3.2 Loop Duration

We measure the loop duration as the time period from the receipt of the routing update until when probes can reach the destination without experiencing loops. Figure 6.8 shows about 70% loops last less than 350 seconds. Note that for loops lasting

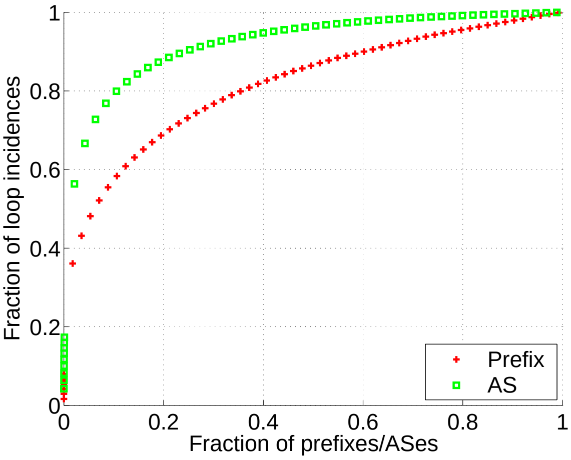


Figure 6.7: Destination prefixes and ASes affected by forwarding loops.

ASN	Loop Incidences	Destination Prefixes	Responsible AS Name	Tier	Primary Country
701	34457 (18.6%)	2059	ALTERNET-AS UUNET Technologies, Inc.	1	United States
1239	33998 (17.7%)	2013	SPRINTLINK Sprint	1	United States
3356	32674 (17.6%)	1998	Level 3 , LLC Communications, LLC	1	United States
7018	27971 (15.1%)	1587	ATT-INTERNET4 AT&T WorldNet Services	1	United States
174	21060 (11.3%)	1149	PSINET PSINet Inc.	1	United States
2914	13612 (7.3%)	787	Verio, Inc.	1	United States
4134	13362 (7.2%)	534	CHINANET-BACKBONE No.31, Jin-rong Street	2	China
6453	13106 (7.0%)	746	TELEGLOBE-AS Teleglobe Inc	1	United States
3549	12267 (6.6%)	850	GBLX Global Crossing	1	United States
3561	12087 (6.5%)	691	CWUSA Cable & Wireless USA	1	United States

Table 6.4: Forwarding loop incidences in the top 10 responsible ASes.

longer than 300 seconds from the first probe, we overcome our maximum probing duration of 300 seconds by background probing to such long-lasting loops to determine whether they are persistent loops. We found that only 0.0027% loop incidences are persistent loops, 74% of which occur close to the destination networks. In addition,

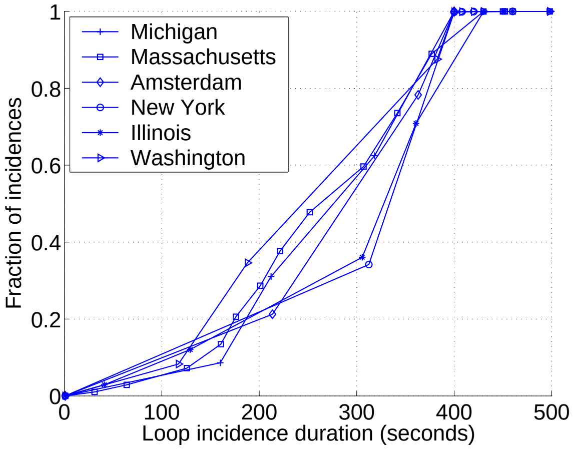


Figure 6.8: Duration of loop incidences.

we observe that the vast majority of loops involves a small number of IP level hops. For example, 81% of loops involve two IP addresses.

6.4.3.3 Loop Predictability

Similarly, we study how predicable loop incidences are. The appearance probability and conditional probability (conditioned on the responsible AS) of loop incidences are shown in Figure 6.9. Only around 20% of prefixes have appearance probability of more than 0.5, indicating that loop incidences are difficult to predict for most prefixes based simply on the presence of any update to the prefix. However, 75% of prefixes have conditional probability (conditioned on responsible AS) of more than 0.5. This illustrates that, given a routing update to the prefix and the responsible AS, loop incidences can be much more predictable.

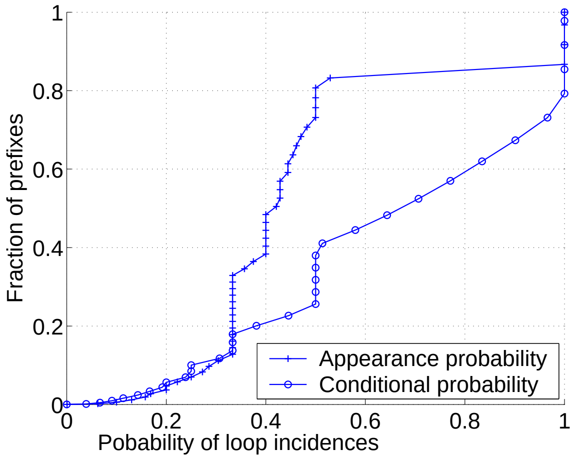


Figure 6.9: Appearance probability and conditional probability (conditioned on the responsible AS) of loop incidences.

6.5 Failure Prediction Model

How well does a routing update indicate the occurrence of a data plane failure? Can we detect the presence of such a failure based on observed routing updates? To answer these questions, we develop a statistical prediction model to infer the probability of a data plane failure given a routing update. As observed in our experiments, the predictability of failure incidences given routing updates across all prefixes follows a bi-modal distribution: some prefixes are highly predictable, while others are not. In this section, we focus on prefixes which are more predictable as analyzed in Section 6.4. We first present our prediction model and then verify the model via supervised learning. Finally, we discuss applications of the prediction model.

6.5.1 Prediction Model

In this section, we derive a model for predicting whether a failure incidence Y occurs upon observing a routing update R to a given destination.

6.5.1.1 Model

We use the random variable Y to represent the data plane observation: $Y = 1$ if there is a failure in the data plane, and $Y = 0$ otherwise. We use the random variable R to represent routing updates with AS path $x_1, \ldots, x_n$. The model is built based on observations of <failure Y, routing update R > pairs in the history data.

In our model, we use a direct acyclic graph (DAG) to represent all the paths for each destination prefix. Each node in the graph represents an AS. In addition, we assume that failures are independent. To determine whether a data plane failure will occur, i.e.$Y = 1$ given a routing update R, we compute the data plane failure likelihood ratio.

$$\Lambda(Y) = \frac{P(Y = 1|R; D)}{P(Y = 0|R; D)} \tag{VI.1}$$

where $P(Y = 1|R; D)$ is the conditional probability of data plane failure occurrence given a routing update R for prefix D, and $P(Y = 0|R; D)$ is the conditional probability of no data plane failure occurrence given a routing update R for prefix D. We say that a data plane failure occurs if $\Lambda(Y) > \lambda$, where λ is a decision threshold which determines false positive and negative rate.

Given an update R with the AS path $x_1, x_2, \ldots, x_n$, if a failure occurs in x_b, then the ASes along the path can be classified to three categories:

- ASes $x_1, \ldots, x_{b-1}$ appearing in the path before x_b are "good" AS nodes;
- AS x_b is a "bad" AS node, also known as the responsible AS.
- ASes $x_{b+1}, \ldots, x_n$ appearing after x_b in the path are "unknown" AS nodes.

Therefore, the probability of AS x_i being a bad node for destination D can be computed as

$$P(Y=1|x_i;D) = \frac{BadCount(x_i)}{TotalCount(x_i)} \tag{VI.2}$$

where $BadCount(x_i)$ is the number of occurrences AS x_i appears as a bad node for destination D, and $TotalCount(x_i)$ is the total number of occurrences AS x_i appears in the path for destination D.

Thus, given a routing update R with AS path $x_1, \ldots, x_n$ for destination D, the probability that R will cause a data plane failure is

$$P(Y=1|R=x_1,x_2,\ldots,x_n;D) = 1 - \prod_{i=1}^{n}(1-P(Y=1|x_i;D)) \tag{VI.3}$$

Similarly, the probability that R will not cause a data plane failure is

$$P(Y=0|R=x_1,x_2,\ldots,x_n;D) = \prod_{i=1}^{n}(1-P(Y=1|x_i;D) \tag{VI.4}$$

After computing the failure likelihood ratio $\Lambda(Y)$, we use the receiver operating characteristic (ROC) in signal detection theory [49] to decide the value of λ. ROC curves are commonly used to evaluate prediction results. In particular, the ROC of a predictor shows the trade-off between selectivity and sensitivity. A curve of false positives ratio (false alarms) versus true positive ratio (detection accuracy) is plotted while varying a sensitivity or threshold parameter. In our experiment, given $\Lambda(Y)$, we determine the ratio of false positive, P_{FP}, and the ratio of detection accuracy P_{AC}, with varying values of λ.

6.5.1.2 Validation

We evaluate both false positive ratio and false negative ratio of our prediction model. A false positive refers to the case where our prediction model predicts a data plane failure given a routing update, while there is no failure observed in our experiment. A false negative refers to the case where our prediction model fails to predict a data plane failure given a routing update. As we have observed in Figures 6.6 and 6.9, some prefixes are more predictable than others. The poor predictability on certain prefixes could be explained by inherent non-stationary properties associated with certain failures, or by the limited visibility from the vantage points of our experiments. Next, we analyze the predictability across different prefixes and focus on the set of more predictable prefixes to further evaluate our prediction model.

We repeat the following experiments 10 times. We first divide the data set into the training set and the testing set. In particular, we randomly sample 50% of the entire observations as the training set and compute the failure likelihood ratio for all the routing updates in the test set. During the training process, we only consider observations that appear at least k times for a given prefix and a responsible AS.

In our experiment, we choose $k = 3$. As a result, we discard 5.6% of observations. Using $k = 4$ will increase the prediction accuracy by 0.34%, while discarding 1.3% of observations.

Next, we compute the average $\Lambda(Y)$ of each prefix for both unreachable and reachable incidences based on our observations. Figure 6.10 shows that the prediction accuracy is limited considering all observed prefixes. Given $\lambda = 1$, 61% of prefixes are predictable (i.e.$\Lambda > 1$) for all failure incidences and 72% of prefixes are predictable (i.e.$\Lambda < 1$) for all non-failure incidences.

Given a prefix and its identified responsible AS, we identify over 51.2%-54.3% of probed updates to be predictable for causing reachability loss and 48.9%-57.5% for causing loops across all six vantage points. The corresponding figures for probed prefixes are 58.7%-67.5% and 53.2%-55.8%, respectively. These destination prefixes account for 28.1%-32.4% of announced prefixes originating from 27.4%-31.9% of all ASes. 3.8% and 5.1% of such destination ASes are tier-1 and tier-2 ASes respectively. The figures for tier-3, tier-4, and tier-5 ASes are 22.4%, 23.6%, and 45.1%, respectively. The set of responsible ASes for unreachable and loop incidences consists of 10.8% tier-1 ASes, 11.9% tier-2 ASes, 19.7% tier-3 ASes, 21.2% tier-4 ASes, and 36.4% tier-5 ASes. This shows that prefixes from the edge of the Internet are more predictable and most responsible ASes are also from the edge.

Figure 6.11 shows the receiver operating characteristics curve of predicting the incidences in the test set. The false positive ratio is shown in x-axis and the prediction accuracy ratio is shown in y-axis. We observe that, by varying λ, our prediction model achieves different degrees of accuracy. For example, with $\lambda = 1$ (i.e.if $\Lambda(Y) > 1$,

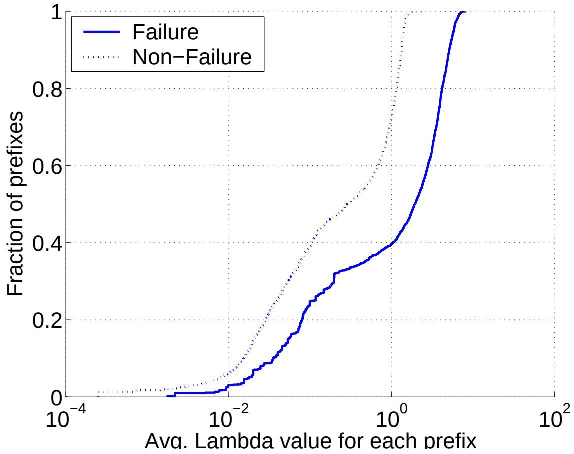


Figure 6.10: Average value of Λ for each prefix

failure is predicted to occur), our model can achieve 89.8% prediction accuracy with 14.5% false positives for unreachable failures and 79.9% accuracy with 12.3% false positives for loops on the subset of prefixes selected above. This observation implies that the prediction model built on history observation can be used to predict future failures on certain prefixes. Figure 6.12 shows the corresponding curves for all prefixes. Given $\lambda = 1$, our model achieves 51% and 60% prediction accuracy for unreachable failures and loops with false positive ratio of 21% and 18%, respectively. This is consistent with our observation in Figure 6.10 that the predictability in general is limited. However, compared to existing work [29] on predicting data plane performance degradations with only 50% prediction accuracy with 60% false positives, our model is much more accurate.

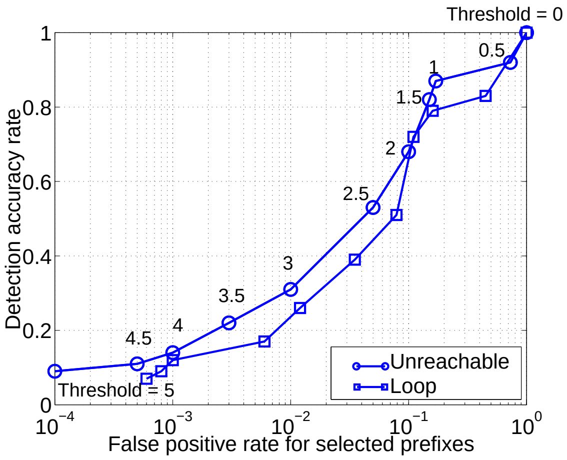


Figure 6.11: Receiver operating characteristics for selected subset of prefixes.

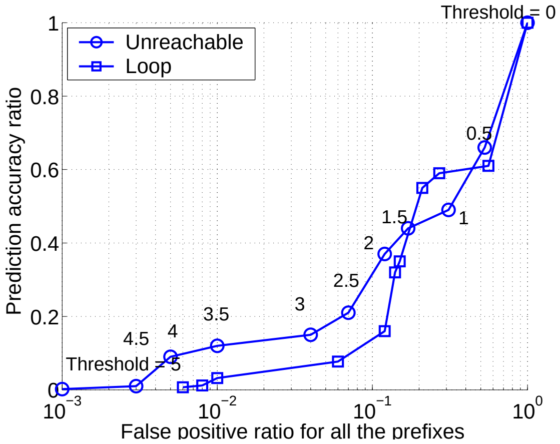


Figure 6.12: Receiver operating characteristics for all probed prefixes.

6.6 Summary

In this chapter, we develop an efficient framework to measure and predict data plane performance degradation as a result of routing changes. To achieve a comprehensive characterization of many diverse routing changes, we develop an efficient and novel measurement framework deployed at each vantage point with access to real-time BGP routing updates. Light-weight probing is triggered by locally observed routing updates. The probing target is an identified live IP address within the prefix associated with the routing change.

Using this framework, we conducted a large scale Internet measurement study and characterized data plane performance upon receiving a BGP routing update. Our experiments and analysis cover a large portion of the announced prefixes and ASes on the Internet. We observe that the data plane performance of a certain set of prefixes is highly predictable. Analyzing the probing results from the framework, we present various properties of the failures across different vantage points. Based on identified inherent stability of routing changes, we analyze the predictability of observed performance degradation from the routing updates. We further develop a statistical model which can accurately predict the severity of potential data plane failures based on observations of routing updates for a given prefix. The ability to accurately predict routing-induced data plane failures is directly useful for applications such as overlay route selection and backup path selection. We show that our model is very useful in a number of applications such as route selection in an overlay network. The model demonstrates the capability for end users to construct short-term mitigation with the assistance of monitoring and diagnosis systems.

Finally, we will present another application to show its long-term benefits of end-host based monitoring system, i.e.comparing ISPs in the long term with various performance metrics. Accurately monitoring ISP performance is important for customers to make informed decisions on ISP selection using a customized ranking of performance metrics of interest. On the other hand, it also provides incentives for ISPs to closely monitor its network and improve the performance to attract more customers.

Systems proposed in Chapter IV, Chapter V, and Chapter VI focus on locating the cause of end-to-end path performance disruptions. In Chapter VII, we will propose a system that can accurately and scalably compare across core ISPs at the same time from end-user's perspectives using multiple fine-grained performance metrics of interest.

CHAPTER VII

Comparing Backbone ISPs using Multiple Performance Metrics

7.1 Introduction

The quality of all network application services running on today's Internet heavily depends on the performance assurance offered by the Internet Service Providers (ISPs). In particular, network providers inside the core of the Internet are instrumental in determining the network properties of their transit services due to their wide-area coverage, especially in the presence of the increasingly deployed real-time sensitive network applications such as financial transactions, video conference calls, and remote surgeries. End-users in the consumer community interact closely with broadband ISPs such as Comcast and RoadRunner. However, organizations such as universities, businesses, or small network providers are direct customers of large core ISPs such as AT&T and Sprint. Accurately monitoring ISP performance is important for customers to make informed decisions on ISP selection using a customized ranking of performance metrics of interest. On the other hand, it also provides incentives for ISPs to closely monitor its network and improve the performance to attract more customers. In spite of such critical importance, there is unfortunately no easy way to effectively compare the quality of network services offered by different core ISPs today. We focus on the services offered by the core ISPs as opposed to broadband ISPs in this work, both due to their larger importance as well as more challenges to address the large coverage requirement.

Even though network transit services are of high importance to business users, it is rather opaque in terms of how to evaluate performance of different core ISPs. Service Level Agreements (SLAs) are the only existing ways for quantitatively comparing across ISPs and can only provide limited, coarse-grained measures for interpreting the services offered. Usually such SLAs are in the form of average values over monthly durations. It is unclear how such SLAs can be effectively verified to ensure ISPs' close compliance to their promised contracts. Besides these existing SLAs [3, 7, 9, 12], there is however no comprehensive and standard way of comparing ISPs using the metrics that end-users truly care about, e.g.fine-grained measures such as packet loss rate, delay, bandwidth, availability, and stability. There exists limited commercial systems, e.g.Keynote, to monitor the latency and loss rate of ISP internal paths. It requires the measurement equipment to be collocated in all ISP's major PoPs, for which ISP cooperation is necessary and is thus challenging to deploy. Moreover, ISPs can intentionally mislead the comparison result produced by such a system given the locations of measurement boxes are known. The closest work on monitoring ISP performance using end-host based probing is NetDiff [83] which measures only latency for only one ISP at a time. However, monitoring a single metric is usually insufficient as application performance may be affected by multiple factors. In addition, the lack of concurrent monitoring of multiple ISPs may lead to unfair comparison.

We designed a system that can accurately and scalably compare across core ISPs at the same time from end-user's perspectives using multiple fine-grained performance metrics of interest. These performance metrics are directly relevant to network applications. We emphasize the importance of examining multiple metrics as different applications have heterogeneous requirements on the network depending on their specific real-time QoS requirements.

In this work, we present our prototype called NetCmp that monitors multiple metrics on multiple ISPs deployed on the Planetlab testbed. NetCmp can simultaneously perform both long-term and instantaneous real-time comparisons across multiple ISPs using key performance metrics such as loss and reachability. We deployed NetCmp to continuously monitor 19 ISPs simultaneously for one month for comparing them under several dimensions, including previously overlooked aspects in SLA definitions such as consistencies in different geographic regions and temporal stability. By correlating multiple metrics, we clearly demonstrate the necessity for comparing ISPs under different metrics to assist more informed provider selection or traffic engineering. Our system runs solely at end-hosts without requiring any proprietary internal ISP data nor special support from the network, and therefore accurately reflect the performance experienced from the perspectives of end systems. Our work provides a solid foundation for developing performance benchmarks comparing QoS metrics across major ISPs. Using NetCmp, we found the network performance measured differs significantly across ISPs. The relative ranking of each ISP under different metrics can vary to a large degree as well. For instance, some ISPs with small internal latency can have many more unreachability events. Finally, although the ranking is stable in short term period, (e.g.in a few weeks), it can change across several months, suggesting the necessity for continuous ISP monitoring, which our system is designed for.

The chapter is organized as follows. We first present our measurement methodology and data analysis techniques to carefully address measurement artifacts in §7.2. Experimental setup and validation of our measurement methodology are presented in §7.3 and §7.4 respectively. Analysis of experimental results of monitoring 19 ISPs for one month is described in §7.5.

7.2 Methodology

A backbone ISP typically comprises of a number of PoPs (Points of Presence) at different geographical locations, inter-connected with dedicated high-speed links. Its primary job is to carry traffic from an ingress PoP to a destination IP prefix. The performance of the traffic depends on the conditions of the *internal path* from the ingress to the egress as well as the *destination path* from the ingress/egress to the destination. We view the ISP as a collection of destination paths and internal paths and use the aggregate performance of all the individual paths as a measure of the overall ISP performance.

In the following, we elaborate on our methodology for measuring the performance of an individual path, including what metrics to measure, how to aggregate samples over time, and how to deal with noise in the data. In §7.5, we will provide concrete examples of ISP performance over particular sets of paths that are relevant to customers.

7.2.1 Path metrics

For each path, NetCmp measures four basic metrics: unreachability, latency, loss rate, and diversity. The first three metrics have direct impact on the traffic performance perceived by customers. The last one is normally transparent to customers, but may indirectly affect unreachability. NetCmp measures each path metric using traceroute-like probes from end hosts. To probe a particular hop on a path, it sends probe packets with pre-computed TTL (time-to-live) value which is expected to trigger ICMP time exceeded message from that hop. Depending on the metric to measure, NetCmp may probe a subset of or all the hops along a path.

7.2.1.1 Unreachability

If any traffic traversing a router fails to reach a destination, the router is considered to experience an unreachability event, i.e., reachability loss, to that destination. Unreachability events can be caused by problems such as network failures, router misconfigurations, and traffic filtering. It is is one of the most essential metrics for assessing the quality of transit service provided by ISPs, as reachability loss is one of the worst-case scenarios for impacted customers. By comparing the unreachability behavior across multiple ISPs, NetCmp provides insights on which ISPs are the most reliable in terms of delivering traffic to certain destinations, to what extent the unreachability of the PoPs within the same ISP differs, and which ISPs are affected by the same unreachability events at the same time. Based on the answers to these questions, customers may select one or multiple ISPs to meet the reliability requirements of their applications.

It is straightforward to measure unreachability from a PoP to a destination given a traceroute path that traverses the PoP to that destination. We consider the destination unreachable from the PoP if traceroute probing stops after reaching the PoP but before reaching the destination. Note that this simple definition can produce misleading results due to several types of noise described below which we carefully address.

First, traffic filtering can make certain destinations seemingly unreachable. Such filtering common in routers and firewalls to block unwanted traffic for security considerations. Routers may also be configured not to generate any ICMP response messages to prevent overloading router processors. Second, routing on the Internet is often asymmetric. Thus, failures on the reverse path from the destination to the probing source may lead to the loss of ICMP response messages, without affecting the reachability on the forward path. Third, probe packets and ICMP response packets may be dropped due to congestion on either direction. Many routers also limit the rate of generating ICMP messages, making them temporarily unresponsive to probe packets.

We deal with these uncertainties as follows. For a given PoP P and destination d, NetCmp maintains a history of whether d has ever been reached from P in the entire probing period. If d is reachable from P previously but becomes unreachable at a particular time t, it is considered a possible unreachability event. Our assumption is traffic filtering policies are unlikely to change frequently (at least in a few weeks), given that they are often manually configured by network administrators.

To eliminate ambiguity that arises due to reverse path failures, NetCmp confirms an unreachability event only if d cannot be reached from P by at least R distinct probing sources. Since the reverse paths from d to different probing sources are likely different with limited sharing, the unreachability event is more likely caused by a failure on the forward path from P to d which is shared by all the probing sources. The choice of R reflects the trade-off between accuracy and coverage. A larger R leads to higher confidence in determining forward path unreachability, but reduces the number of (P, d) paths we can study, given only a limited number of probing sources and with only a subset of them traversing (P, d). We have tried a few values of R and found $R = 3$ achieves a reasonable balance between the two conflicting goals.

To reduce the impact of packet loss caused by network congestion and router rate-limiting, NetCmp will probe a hop at least 3 times with an interval of I seconds between two consecutive probes. I should eclipse most congestion periods and be large enough to avoid triggering rate-limiting on most routers. In the current system, we choose I to be 1 second.

7.2.1.2 Loss rate and latency

Loss rate and latency are the two of the most important metrics in evaluating the performance of ISP networks. They directly affect the performance of most Internet applications. NetCmp monitors these two metrics of multiple ISPs simultaneously. Customers may select ISPs based on the values of these two metrics and the QoS (Quality of Service) requirements of the applications, e.g.sending delay-sensitive traffic to ISPs with low latency. ISPs may also use our measurement results to compare with the performance of their competitors' networks, to identify bottlenecks within their own networks for improving service quality.

Given an internal path or a destination path, NetCmp measures its loss rate by probing the two ends of the path from a given probe source. This means it will only probe the hops corresponding to an ingress, an egress, or a destination instead of all the hops along the path to reduce overhead. To measure the loss rate to a particular hop, NetCmp sends probe packets with pre-computed TTL values expected to trigger ICMP time exceeded response from that hop. It probes each hop 200 times to detect loss rate as low as 0.5%. Sending more probes to a hop increases the sensitivity of loss rate detection but also the probing overhead. The loss rate of the path is computed by subtracting the loss rate of the starting hop from that of the ending hop. To avoid triggering IDS (intrusion detection system) alarms from edge networks, NetCmp currently only measures the loss rate of ISP internal paths.

There are a few factors that may lead to inaccuracy in our loss rate measurements. First, NetCmp uses single-ended probes to measure forward path loss rate. It may over-estimate the actual loss rate given that it cannot distinguish forward path loss from reverse path loss. Past work suggests that big packets are more likely to be dropped than small packets [79, 81]. Since ICMP response packets are relatively small (56 bytes), we use 1000 byte probe packets to ensure the measured loss is mostly on the forward path. Note that this also biases our loss rate measurements towards loss experienced by relatively large packets. Second, end hosts have limited CPU and network resources, making our measurement results susceptible to resource contention on the probing hosts, in particular Planetlab hosts [108]. To overcome this type of interference, NetCmp continually monitors the load on each probing host and filters out abnormal loss rate samples that co-occur with high CPU utilization on the host. Third, probe packets and ICMP response packets may be dropped due to router rate-limiting. As described before, NetCmp uses a one-second interval between two consecutive probes to mitigate this problem.

NetCmp applies the latency measurement methodology in NetDiff to measure multiple ISPs concurrently. This is similar to loss rate measurement except that a hop is probed only once to obtain an RTT (round-trip time) estimate. The latency of the path is computed as the half of the RTT difference between that to the starting hop and that to the ending hop. Since the measurement is lightweight, NetCmp measures latencies for both internal paths and for destination paths. The latency of a path is inherently correlated with the direct geographic distance of the path, e.g.long paths tend to have large latencies. To properly compare latencies, we account for the bias introduced by the difference in path distance. We define the *stretch* of a path as the additional latency compared to that of a hypothetical direct-link between its two end points. For instance, if the latency is 50 ms for a path whose direct-link latency is 40 ms (computed using its direct geographic distance and the speed-of-light in fiber), the stretch of the path is 10 ms. In the remainder of this chapter, we will use stretch instead of absolute latency for ISP performance comparison.

Since NetCmp infers forward path latencies based on RTT measurements, assuming equal contribution from both forward and reverse path latencies. The results can be distorted by path asymmetry. To guard against such errors, NetCmp will discard RTT samples of any hop whose forward path hop count and reverse path hop count differ by more than three. The latency estimates can also be distorted by the heavy load on a probing host, which we detect by capturing the variance of the RTT samples of a particular hop. Samples associated with abnormally large variance are discarded [83]. Note that these heuristics are limited and thus may include samples affected by path asymmetry or exclude samples of actual large latency fluctuations. But for the purpose of comparing ISPs using latency as a metric, we believe this approach as used in NetDiff is reasonable.

7.2.1.3 Diversity

Given a PoP P and a destination d, we define the *diversity* of path (P, d) as the total number of distinct AS paths from P to d. Unlike the previous three metrics, path diversity is not directly visible to the ISP customers. But it provides insight into an ISP's ability to find alternate paths to bypass failures on the default path. The failure location also plays a role, as failures near edge networks are more difficult to overcome compared to those inside core networks where alternate paths are abundant.

NetCmp measures the diversity of a path (P, d) by periodically conducting traceroutes that traverse (P, d). The diversity is estimated as the total number of distinct AS paths from P to d observed during the entire measurement period. This is a coarse-grained measure as previously observed paths may no longer be usable. We convert IP paths measured by traceroute into AS paths using IP-to-AS mapping. An IP address is mapped to ASes based on its origin ASes in the BGP tables. One IP address may map to multiple origin ASes (MOAS) and we keep a set of origin ASes for such IP addresses.

Some inferred AS paths may contain AS loops. While temporary loops may arise during routing convergence [60], persistent loops are disallowed by BGP and should rarely occur. They usually indicate incorrect IP-to-AS assignments and are thus excluded from our study. Due to limited CPU and network resources at probing hosts, NetCmp cannot probe each (P, d) as frequently as desired. This means it may miss certain AS paths that appear only between two consecutive probes, leading to an underestimate of diversity.

7.2.2 Path scores

Individual unreachability, latency, or loss rate sample reflects the instantaneous performance of a path. They are informative in predicting the short-term path performance and optimizing route selection. Nonetheless, individual samples capture the transient behavior of networks, which may change over time. This makes them less useful in helping customers select the best ISPs based on long-term projection. Past work [136] has demonstrated the feasibility of improving end-to-end performance and reliability using frequent active probing. We develop temporally stable measures that can represent the overall ISP performance.

NetCmp periodically monitors the performance of each path. By aggregating the samples over time, it obtains a long-term, average *score* of path performance. In

§7.5, we will demonstrate such scores not only capture the inherent performance difference between ISPs but also remain relatively stable over the duration of a few weeks to even a few months. For each metric studied (except diversity), we compute the path score by taking the average of all the samples of the same metric collected on the same path during the entire measurement period. An unreachability score of (P, d) represents the fraction of time when destination prefix d is unreachable from PoP P. Average latency and loss rate are commonly used in the SLA (service-level agreement) specifications of many backbone ISPs [3, 7, 9, 12]. Other aggregation methods, e.g.geometric mean, median, or 90^{th} percentile, can also be used. Due to space constraints, we only present the results based on average in this chapter.

7.2.3 Discussion

NetCmp relies on single-ended probes from end hosts to measure ISP performance. We now summarize the potential sources of noise and describe ways to overcome them. A small fraction (0.18%) of traceroute paths contain transient IP-level loops. We discard such traceroute paths since they are most likely measured during routing convergence period.

Built on past work on inferring ISP topologies [110], we map an IP address to an ISP PoP using its DNS name. Mapping error may arise when an IP address has an incorrect DNS name, often manifesting itself as persistent PoP-level "loops" in the traceroute data. Under normal conditions, traffic that traverses a PoP should not return to the same POP again, because ISPs want to reduce propagation delay and avoid overloading expensive long-haul links. We detect and discard all the suspicious

IP addresses using existing techniques [135].

To compute path stretch from absolute latency, we need to map a destination to a geographic location. The mapping is done with the commercial geolocation database from MaxMind [4]. MaxMind claims 99% mapping accuracy at the country-level and 80% accuracy at the city-level within the USA. To further check for mapping errors, we apply the speed-of-light test on all the paths. The latency of a path inferred from RTT samples should always exceed the minimum time it takes for light to travel between the locations of its two end points. We discard IP addresses that fail in many tests [83].

We have validated our results against various types of noise mentioned above. In §7.4, we use the BGP data to confirm the unreachability and the diversity results. We also thoroughly study the effects of reverse path loss, ICMP rate-limiting, and overloaded probing host on the loss rate measurements to verify the soundness of our methodology.

7.3 Implementation

We implement NetCmp purely using end systems without any ISP cooperation or proprietary data. Our system builds on the probing methodology of Netdiff [83], which is a system for identifying the latency differences between ISP backbones. It measures many paths in one ISP at a time. The main limitations which we improve upon are the single ISP based probing which restricts Netdiff to examine only one ISP at a time, as well as the consideration of only the delay metric which we expanded to reachability, packet loss, stability and diversity metrics.

In the following we briefly summarize the implementation based on the Netdiff framework. It consists of a centralized path selector and a distributed set of probers. It divides the measurement process into cycles and measures a pre-computed set of network paths of a given ISP in each cycle. Each cycle lasts at most 1 hour. At the beginning of each cycle, the path selector takes routing views as input and computes a task list of probing destinations for each prober. The routing views are collected using traceroute measurements from all the probers to all the destination prefixes at a low rate. They are updated daily to keep up with the changes of ISP routing topologies. The path selector implements a greedy algorithm similar to NetDiff. Note that path selection is performed for all target ISPs simultaneously instead just for one at a time. This significantly reduces probing overhead by leveraging the fact that a single probe often traverses multiple target ISPs.

The second major components in NetCmp are a set of distributed end hosts, accepting the destination list from path selector performing the probing. Probing is conducted with a customized version of traceroute that measures multiple hops of a path and multiple destinations in parallel. The probing packets are constructed to reduce the probability that different probing packets from the same source to the same destination take different router-level paths due to load-balancing within the ISP [19].

ISP	Tier	# of PoPs	PoP-dst (x1000)		PoP-PoP
			All	Redundancy(3)	All
Qwest	1	49	980	752 (76%)	716
UUNet		139	2919	2363 (80%)	2059
Sprint		57	2052	1389 (67%)	1213
AOL Transit		25	150	122 (81%)	232
Verio		46	1334	1093 (81%)	511
Level3		71	2556	1993 (77%)	1501
Global Crossing		59	1121	997 (88%)	677
Savvis		38	380	358 (94%)	502
AT&T		112	2456	1705 (69%)	822
XO	2	45	187	159 (85%)	539
British Telecom		32	315	294 (93%)	419
Tiscali		30	151	144 (95%)	325
Deutsche Telekom		64	147	103 (70%)	115
France Telecom		23	138	101 (73%)	202
Broadwing		19	146	102 (69%)	137
Teleglobe		44	242	204 (84%)	539
Cogent		69	1039	822 (79%)	1787
AboveNet	3	44	756	504 (66%)	261
Abilene		11	96	71 (73%)	49

Table 7.1: Data collection (All: all measured paths; Redundancy(3): paths traversed by at least 3 different sources).

7.4 System Evaluation

We have been running NetCmp for more one month on the PlanetLab distributed testbed [101] to study 19 ISPs in parallel. We set up the system on 750 hosts in about 300 distinct sites. The results in this work are computed from the data collected among 19 ISPs for 30 days, from Apr. 26 to May 9 and from Nov. 2 to Nov. 16 2008. We plan to use NetCmp to continuously monitor ISPs' performance in the long term and publish our findings on the Web. As shown in Table 7.1, NetCmp covers all the major PoPs of the 19 ISPs and most of the internal PoP level paths (validated using the Rocketfuel data [110]). In this section, we describe our experiment setup

ISP	PoPs with for BGP feeds	PoP-dst coverage	Detected unreachable events	Distinct paths confirmed w BGP
UUNet	NewYork	9468	38 (80%, 76%)	40712 (99% 98%)
Sprint	Pennsauken	15038	89 (79%, 89%)	49623 (81%, 87%)
AOL	Chicago	2789	19 (81%, 76%)	8378 (95%, 98%)
Verio	Ashburan,London	13924	75 (90%, 78%)	29240 (95%, 97%)
Level3	Denver,London	20608	135 (82%, 90%)	30919 (99%, 98%)
GlobalX	NewYork	12810	76 (91%, 82%)	32015 (100%, 96%)
Savvis	SF,Sunnyvale	3339	15 (75%, 53%)	7345 (90%, 93%)
ATT	NewYork	27839	129 (90%, 86%)	41758 (90%, 96%)
BT	London	5012	47 (82%, 78%)	9021 (93%, 96%)
Tiscali	SF,London	5539	20 (78%, 66%)	17170 (91%, 98%)
FranceT	NewYork,London	2237	42 (88%, 84%)	5145 (100%, 96%)
Teleglobe	PaloAlto,London	7797	69 (86%, 87%)	24955 (95%, 93%)
Abilene	NewYork, Chicago, *etc.*	13927	35 (94%, 77%)	52922 (95%, 97%)

Table 7.2: Validation using BGP data for route instability and unreachability events.

and validation of our measurement methodology.

Table 7.1 shows the overall coverage of the ISP PoPs in column 3, the PoP-destinations in column 4, and the internal paths in column 6, as observed from our data set. In order to accurately detect reachability problems, for ISP PoP-destination path, we require a sufficient number of paths traversing it. In our study, we focus on detecting reachability of a subset of paths that are probed from at least three different hosts to balance the trade-offs between redundant coverage and probe overhead, given the limited resource available on each prober. As is shown in column 5, the filtering process only eliminates less than 20% paths for most ISPs. Monitoring all 19 ISPs simultaneously, it takes about *an hour* for all the hosts to finish the scheduled probe tasks. This limits us to only focus on long-term, average performance of the ISPs. For instance, the transient reachability problems or short-lived congestion events may be missed by NetCmp. However, we believe long-term performance metrics are more representative and stable for cross-ISP comparison.

7.4.1 Validation

We first evaluate the accuracy of reachability event detection using public BGP feeds from RouteViews, RIPE, and Abilene. For most ISPs from which we have BGP feeds, we only have the data from at most two routers. For each ISP, we first identify the corresponding PoP where the BGP feed comes from shown in the third column in Table 7.2. As expected they are mostly from PoPs in major metropolitan areas. As an exception, for the educational backbone Abilene network, we have access to BGP feeds from all its PoPs. Because different PoPs in an AS can experience distinct routing changes due to different external BGP peerings, we compare BGP-observed changes with traceroute-observed changes only when our probes traverse the PoP where the BGP feed comes from. Note that we do assume consistent routing within the same PoP, as such routers are physically co-located and usually configured with the same routing policies.

Due to the limited number of probing hosts and their limited network locations, we cannot cover all the paths to any destinations from every ISP at each location with a BGP feed. We first identify the set of destinations whose paths go through the corresponding PoP with BGP feed for each ISP, as shown in the third column of Table 7.2. The coverage varies quite significantly across ISPs, from 5012 for British Telecom in London to 27839 for AT&T in New York city. Large ISPs and US-based ones naturally have better coverage due to higher chance of traversing such networks given the prevalent U.S. location of PlanetLab nodes. Coverage is not a fundamental limitation to our work. With more probing hosts becoming available, coverage can be improved.

For the subset of destinations which can be used for comparison, we identify any reachability loss events based on BGP withdrawal messages lasting longer than our probing interval from the BGP data. If no other larger prefix covering the withdrawn prefix exists in the table, this prefix is unreachable from this router's perspective from the time the withdrawal is received till the next routing update. Then we check whether there is any reachability loss events reported by NetCmp during the withdrawal period. We calculate the recall and precision of our reported reachability loss events in column four. *Recall* is the fraction of reachability loss events in BGP detected by NetCmp. *Precision* is defined as the percentage of reachability loss events in our results also confirmed by BGP. We can see the precision is very high, indicating that we only miss very few reachability loss events. The recall is not as high, indicating potential false positives in our result. This can arise due to several reasons including legitimate ones: (1) the control plane (BGP data) and the data plane (probed paths) may not always agree due to routing or forwarding anomalies. (2) there may exist potentially inconsistent routing views within a PoP. (3) measurement errors may exist in traceroute, and anomalous BGP announcements are also possible. (4) reverse path reachability loss may occur from the router to the probing source.

We use a similar method to validate the distinct number of AS-level paths from each ISP to each destination. We identify any AS-level path being used longer than our probing interval from the BGP data for each destination. We then compare the distinct AS level paths from the traceroute measurement. We present all the distinct

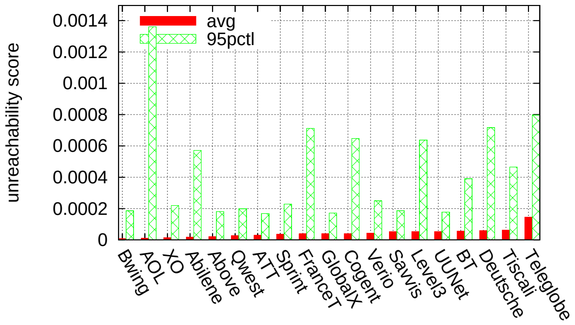


Figure 7.1: Fraction of PoP-dst pairs with unreachability problem.

paths observed during the entire study period, and calculate the recall and precision in a similar way, shown in the last column of Table 7.2, which are both very high. The inaccuracy may be due to errors in IP to AS mapping.

In summary, our detailed and comprehensive validation demonstrate the soundness of our methodology.

7.5 Results

In this section, we present a comprehensive set of the results that compare the performance of the backbone ISPs along multiple dimensions. We first focus on two individual metrics: unreachability and loss rate. For each metric, we compare the overall performance of the ISPs using the samples collected during the entire measurement period on all the relevant paths. Because customers may only be interested in a subset of the paths of an ISP, we perform path-based comparisons tailored for the specific workload requirements of customers. We also study the temporal stability of the comparison results in the short-term and long-term.

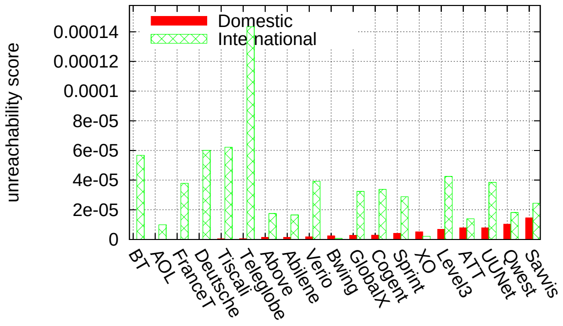


Figure 7.2: Compare unreachability for for U.S. vs. international paths.

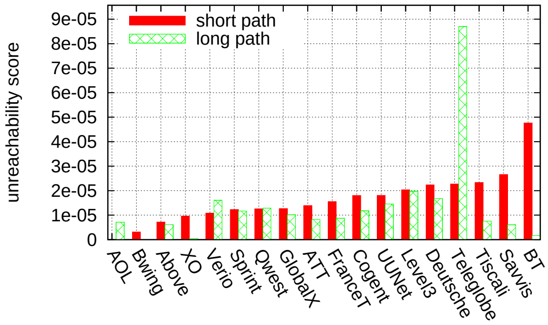


Figure 7.3: Compare unreachability for short paths vs. long paths.

Next, we compare the ISPs using multiple metrics simultaneously. The results of such comparisons are particularly useful for customers who have QoS requirements on more than one metric, e.g.both loss rate and latency. They also help customers understand the trade-off and correlation between different metrics and moreover help ISPs make informed optimization to their networks. The results in this section are not exhaustive but highlight the kinds of insights on ISP performance that NetCmp can provide.

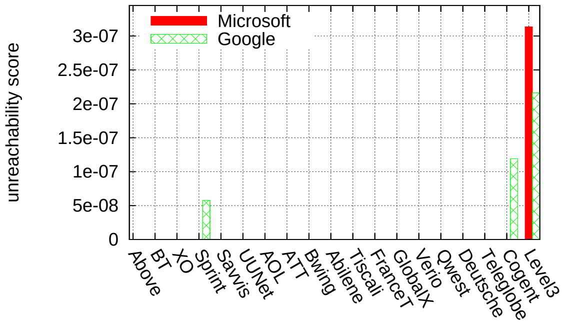


Figure 7.4: Compare unreachability to Microsoft and Google

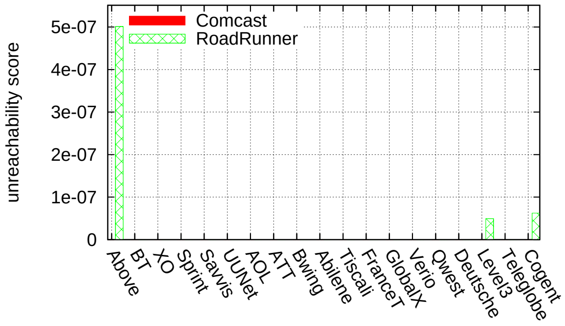


Figure 7.5: Compare unreachability to Comcast and RoadRunner

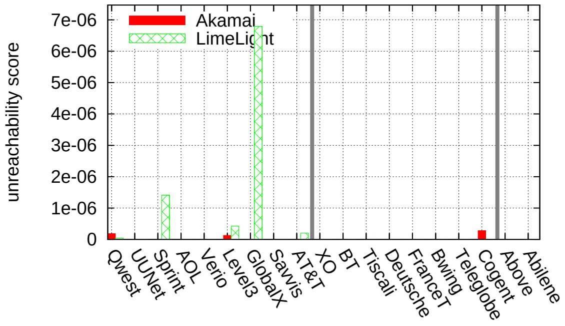


Figure 7.6: Compare unreachability to Akamai and Limelight

7.5.1 Unreachability

Figure 7.1 shows the overall unreachability scores of all the destination paths across different ISPs. The y-axis is the score and the x-axis is a list of ISPs sorted

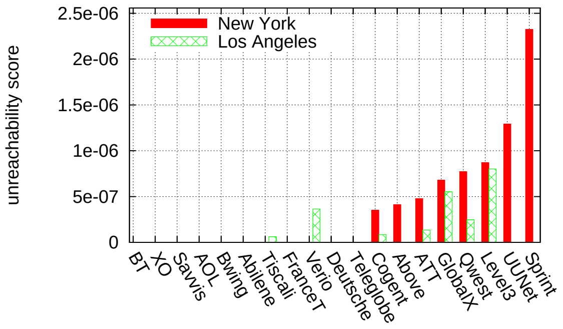


Figure 7.7: Compare unreachability in two PoPs: New York and Los Angeles to prefix 84.38.208.0/20 (a tier-3 U.S. ISP).

based on the scores, from the best to the worst. For each ISP, we compare the average and the 95^{th} percentile of the scores of all the destination paths. On the one hand, unreachability events are rare. For all the ISPs, the availability of a destination from a PoP is over 99.99% on average. On the other hand, the relative difference between ISPs is significant, indicating the importance of ISP selection for mission-critical applications. The average unreachability of Teleglobe is over an order of magnitude larger than that of Bwing. The 95^{th} percentile scores are significantly larger than the average, suggesting unreachability events occur much more frequently to a small set of destinations. Further investigation reveals that many of these problematic destinations are shared among the ISPs, likely because the unreachability events to these destinations affect multiple ISPs at the same time.

7.5.1.1 Path-based comparison

The overall score of an ISP is highly aggregated and may hide many differences that are of interest to particular customers. We now provide a few usage scenarios and study the ranking of the ISPs according to various geographic and network

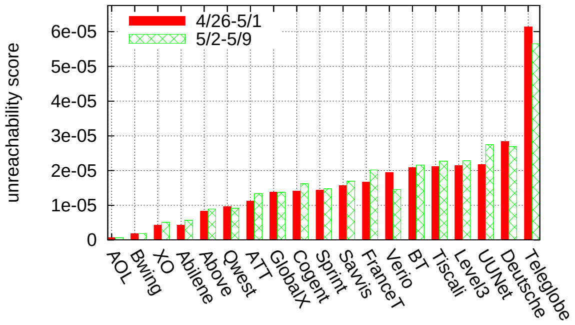


Figure 7.8: Compare unreachability scores for two contiguous time periods.

requirements.

We first study the unreachabilty of the ISPs based on the geographic properties of the paths. Customers may be much more interested in the paths within a specific region where their clients and servers reside. We provide an example to show how NetCmp results may help them choose the appropriate ISPs. Figure 7.2 shows the average unreachability of the domestic and the international paths of each ISP. A path is considered *domestic* if both ends of the path are within the U.S. Otherwise, it is considered *international*. Overall, domestic paths are much more reliable than international paths that often traverse transoceanic links, suggesting the former are better provisioned. The ISP rankings are quite different for the two types of paths. In this example, Teleglobe has pretty good domestic paths but its international paths are highly unreliable. On the contrary, XO ranks much higher with its international paths than with its domestic paths. Thus, the geographic properties of customers' traffic workload may often lead to drastically different decisions on the choice of ISP.

Next, we study the impact of geographic distance on ISP unreachability. We divide paths into the short and the long groups based on their direct-link latency.

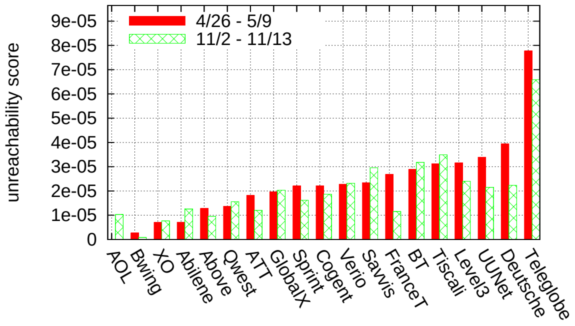


Figure 7.9: Compare unreachability scores for 5-month separated time periods.

Short paths have latency smaller than 40 ms, which usually span one continent. *Long* paths often span across continents. A content provider with world-wide users may be interested in performance for short or long paths or both depending on whether it has presence in every continent or in just one continent. Figure 7.3 shows the average unreachability of short and ong paths of each ISP. There does not seem to be any strong correlation between distance and unreachability. For many ISPs, long paths even have lower unreachability scores. Later, we will show that path diversity has a higher correlation with unreachability. The rankings of the ISPs vary for these two types of paths. While a few ISPs are consistently good or bad, e.g.Bwing and Teleglobe, the relative quality of the others can be quite different. For instance, the ranking of BT is nearly inverted for short and long paths. Therefore, customers should choose ISPs carefully based on required geographic distance traversed by their traffic.

The unreachability of the ISPs heavily depend on the destinations of traffic. Content providers are generally interested in paths to broadband networks with heavy concentration of their users. Similarly, consumers in edge networks usually care about the paths to online service providers like Google, Microsoft, and various CDNs (content distribution networks). Figure 7.4, 7.5, and 7.6 compare the paths to specific destination networks. Compared with Figure 7.1, as expected, the unreachability score of the paths to these popular destinations is much lower than that of the paths to arbitrary destinations. This is likely explained by the fact that popular destination networks are usually well-connected. Except for a few exceptions, e.g.Level3, AboveNet, and GlobalX, most ISPs experience almost no unreachability events to these popular destinations. Such direct comparison provides strong incentive for the ISPs to match the reliability of their competitors.

Finally, we study to what extent the unreachability of the ISPs depends on both locations and destinations. Such comparison results are useful for customers whose traffic always traverses between a city and a specific destination, e.g.VPN connections. Figure 7.7 shows the unreachability of each ISP from Los Angeles or New York to 84.38.208.0/20 (a tier-3 U.S. ISP). Many ISPs are consistently good or bad at reaching the destination from both locations, because they have similar AS paths. Surprisingly, even within the same ISP, there can be significant difference between the reachability from the two locations. This is most evident for Sprint, in which case traffic follows entirely different AS paths from the two locations to the destination as visible in traceroute paths measured.

7.5.1.2 Time-based comparison

We now investigate the temporal stability of unreachability scores. This helps customers understand how quickly they need to change their ISP selection. If the un-

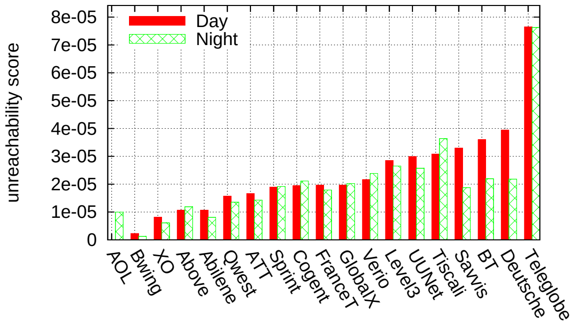


Figure 7.10: Compare unreachability scores between day and night.

reachability scores are too volatile, customers may need to use multi-homing or route selection to optimize the performance of their traffic in a small time-scale. Figure 7.8 shows the unreachability scores of each ISP in two consecutive periods. Clearly, the scores are fairly consistent across all the ISPs, suggesting that the unreachability of the ISPs are stable over a few weeks. Figure 7.9 shows the unreachability scores in two periods with about five months apart. Not surprisingly, the unreachability of the ISPs vary more drastically in these two periods. For instance, France Telecom, Deutsche Telekom, and UUNet all have significant lower unreachability scores. This highlights the importance of continuously monitoring ISP performance to keep up with changing network conditions.

It is well known that traffic on the Internet exhibits diurnal patterns. On the one hand, routers and links are expected to handle heavier loads in the daytime. On the other hand, ISPs are more likely to perform maintenance tasks at night when there is less traffic. In Figure 7.10, we study the time-of-day effect on the unreachability of each ISP. For most ISPs, there is slight difference between the unreachability in the day and the night time. This suggests there is little correlation between network

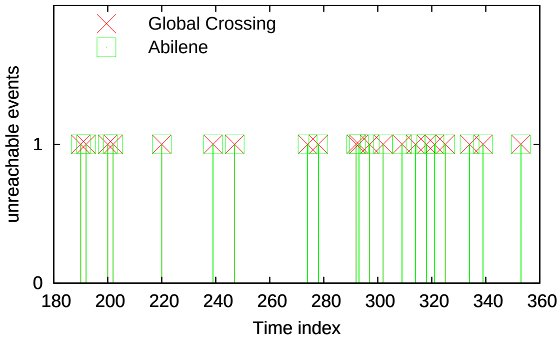


Figure 7.11: Correlation of unreachability to 202.57.3.0/24 (an ISP in Indonesia) from two ISPs: strongly correlated.

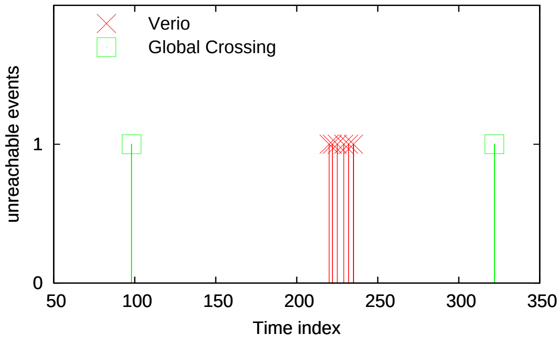


Figure 7.12: Correlation of unreachability to 72.14.235.0/24 (YouTube prefix) from two ISPs: uncorrelated.

load and unreachability. In addition, the maintenance tasks in these ISPs are well planned such that they have minimal impact on the traffic. The unreachability of a few ISPs do vary between day and night. This reminds the customers to consider the worst-case scenario instead of just the average case when choosing ISPs.

7.5.1.3 Time correlation

Failures occurring close to the destination network are likely to affect multiple ISPs simultaneously. On the other hand, failure happening close to the ISP in a single region may only impact one ISP. Here, we give two examples to demonstrate

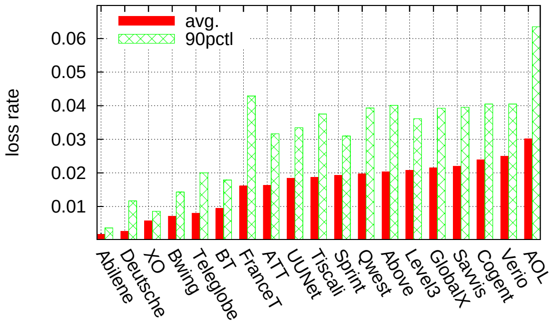


Figure 7.13: Loss rate comparison across ISPs.

these two extreme cases. Figure 7.11 shows that both Global Crossing and Abilene are affected for reaching destination prefix 202.57.3.0/24 (an ISP in Indonesia) at exactly the same time. We confirmed in traceroute that the path stopped in one provider of the destination network, suggesting the failure occurs close to the edge. In this example, even one customer multi-homes to both ISPs, it will still suffer from the failures. Figure 7.12 shows the reachabilty to a highly reliable prefix has little temporal correlation across ISPs. Verio and Global Crossing are likely affected by independent failures far from the destinations. In this example, choosing both ISPs as providers is sufficient to provide full reliability to the destination.

7.5.2 Loss rate

Aggregating all the internal paths of an ISP together, we present the average and the 90^{th} percentile loss rates in Figure 7.13. We observe the loss rate difference between the ISPs is quite big, ranging from less than 0.1% in Abilene to nearly 3% in AOL. Customers of these ISPs are expected to experience significantly different performance, reaffirming the importance of careful ISP selection. The loss rate of

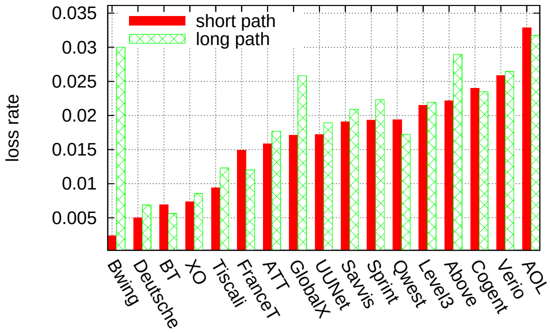


Figure 7.14: Compare average loss rate for long vs. short paths.

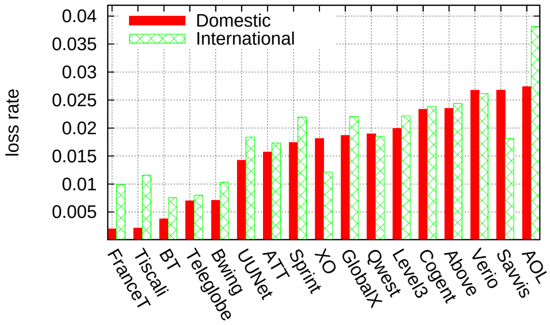


Figure 7.15: Compare average loss rate for U.S. vs. international paths.

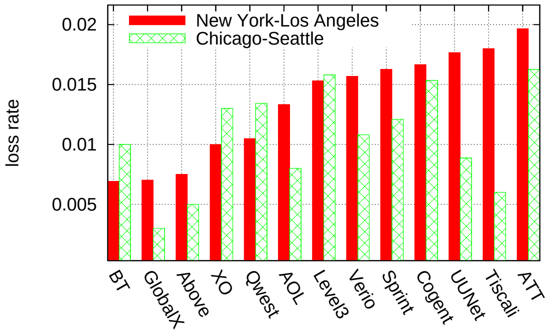


Figure 7.16: Average loss rate distribution between large PoPs.

Abilene is small both for the average and for the 90^{th} percentile, indicating it is well-provisioned for its traffic demand as the educational backbone. Some ISPs rank consistently high or low both for the average and for the 90^{th} percentile, while others have much higher 90^{th} percentile loss rate. For instance, although France Telecom has relatively low average loss rate, there exists a small set of paths with high loss rate. Identifying such lossy paths help ISPs better provision their networks and also help customers make performance-aware routing decisions.

7.5.2.1 Path-based comparison

To reveal the detailed difference across ISPs based on the types of paths using various attributes, we group paths in different ways for further comparison.

Similar to previous reachability analysis, we group all the paths into short and long paths according to their hypothetical direct links. Paths across larger regions, e.g.between two coasts in U.S. are considered long, whereas paths within the same region are considered short. Figure 7.14 shows a lack of any strong correlation between path distance and average loss rate for most ISPs, indicating roughly the same degree of provisioning within the ISP in general. Exceptions do exist, however, for some ISPs, e.g.Broadwing where longer paths have significantly higher loss rate. This implies that customers may choose different providers depending on whether traffic is expected to traverse longer distance. It also informs these ISPs on which paths require performance improvement using techniques such as better load balancing or additional capacity provisioning.

We also categorize all the paths into US based vs. foreign country based sets relying on IP to geolocation mappings. Comparing across these two sets, as expected, Figure 7.15 shows that paths within non-US regions tend to have slightly higher loss rate for most ISPs, given they are mostly US-based ISPs.

Given that there is usually higher traffic demand across PoP pairs in metropolitan areas, we examine the average loss rate of two long paths for two pairs of big cities. Figure 7.16 shows that a given ISP does not always have lower loss rate for both paths. In some cases, an ISP may even have lower loss rate for the longer path (New York-Los Angeles). Such differences are largely due to different internal topology and congestion level. Compared to the average loss values shown in Figure 7.13, paths between big cities are not necessarily less lossy, likely due to high traffic demand for such paths. Moreover, only a few ISPs exhibit low loss for both paths, e.g.Global Crossing. Many ISPs, e.g.Tiscali or XO, differ quite significantly in the average loss behavior for these two paths. This observation suggests that customers may select different ISPs based on the peering location and expected traffic patterns.

7.5.2.2 Time-based comparison

We examine the stability of the loss rate metric in characterizing ISP performance. Figure 7.18 shows average loss rate measured for two distinct contiguous short-term time periods are very consistent, giving the same relative ranking for these ISPs. For long-term comparison shown in Figure 7.17, the two longer time periods separated further apart by 5 months also correlates well except for a few ISPs such as XO and AOL. This inconsistency suggests the usefulness of NetCmp for long-term, continuous ISP monitoring.

The main factor causing packet loss is likely transient congestion. It is known that Internet traffic distribution has time-of-day effects; therefore, we compare average loss rate in each ISP for PoPs within U.S. measured during day vs. that measured at night. In Figure 7.19 we observe for most ISPs the loss rate during day time is only slightly higher than that measured at night, although some exceptions do exist, e.g.AOL, France Telecom. This is likely a result of the ISPs monitored being large ISPs with customers all over the world, leading to less pronounced time-of-day effect. For small ISPs, e.g.Abilene, loss rate during day time is indeed significantly higher. Such observations help customers resort to solutions such as dynamic route selection.

7.5.2.3 Temporal correlation

Large ISPs usually have PoPs co-located at the same exchange points. Thus, the shared underlying physical infrastructure may cause correlated performance degradation among ISPs. We study such temporal correlation across ISPs using the following example. Let us assume a financial institution needs to reliably synchronize data in real time between two data-centers in New York and Chicago. It wants to choose an ISP with low loss rate. Since reliability is a concern, it also wants to use another ISP as backup in case the loss rate of its primary ISP becomes unacceptable. Ideally it should select two ISPs with as independent path performance as possible. Figure 7.20 shows the time series of the loss rate of AT&T and Level3 between Chicago and New York. While the loss rate of both ISPs is reasonably small in most time, they are highly correlated. This could become a serious problem for the institution if the path performance of both ISPs degrades simultaneously. The loss rate spike between interval 120 to 140 is the exact type of scenario that the institution attempts to avoid. This example highlights the utility of NetCmp in helping customers make intelligent decision in selecting ISPs.

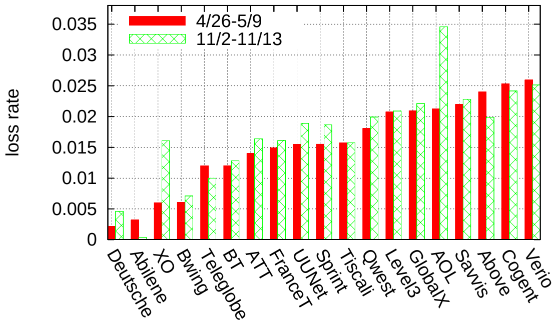


Figure 7.17: Compare average loss rate for 5-month separated time periods.

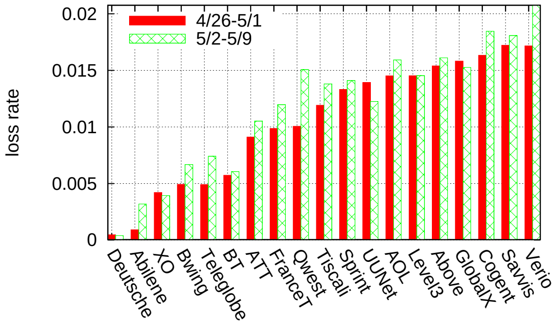


Figure 7.18: Compare average loss rate for two contiguous time periods.

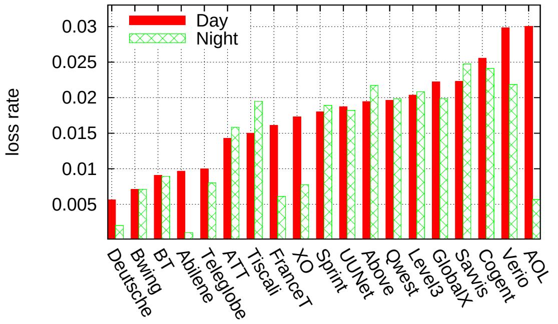


Figure 7.19: Time of day effect on average loss rate.

7.5.3 Correlation between metrics

Anomalous network events, such as routing failures and malicious attacks, are known to introduce large impact on the Internet. These events may be reflected in

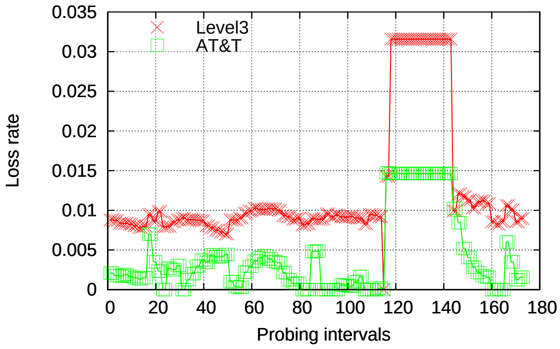


Figure 7.20: Correlation of average loss rate between Level3 and AT&T (from New York to Chicago)

multiple metrics and in different networks. Given NetCmp continuously measures the delay, loss rate, and unreachability of 19 ISPs simultaneously, it offers us a unique opportunity to reveal the performance correlations across metrics and ISPs. These correlations provide a comprehensive comparison across ISPs, helping customers select ISPs to meet the heterogeneous requirements of their applications and to be resilient to network disruptions.

In this section, we provide a few concrete scenarios to highlight the kind of correlation analysis enabled by NetCmp and their use. These scenarios are by no means exhaustive; customers may come up with different scenarios based on their own needs. Furthermore, for customers making decisions to choose ISPs, the ultimate of choice of one or more ISPs can be made using a variety of considerations facilitated by information provided by NetCmp.

7.5.3.1 Correlation between loss and latency

Suppose a large enterprise requires real-time streaming of video traffic between two branch offices in New York and Seattle. It desired to select an ISP that offers network paths with both low latency and loss rate in order to guarantee good streaming quality. To illustrate how NetCmp can help, we plot the ranking of the 13 ISPs that provide transit service between the two cities using both latency and loss rate. As a fair comparison, similar to NetDiff, we use stretch instead of actual latency to eliminate the artifact caused by different ISP size. Figure 7.21 shows the score of loss rate in left y-axis and stretch in right y-axis. We note that different metrics of the same ISP may not be correlated. For instance, Sprint ranks poorly using the loss rate metric but ranks pretty well under the stretch metric. Nevertheless, some ISPs rank consistently high (e.g.Bwing) or low (e.g.Level 3 or Quest). In this case, Broadwing is the clear winner.

7.5.3.2 Correlation between reachability and latency

In our second example, let's assume traders at one site of a bank need to communicate with the server located in 202.65.134.0/24 (a company in Hong Kong) for timely financial transactions. In this scenario, both the latency and the reachability are critical metrics in ISP selection. The traders can use NetCmp to compare the performance of 13 ISPs to this destination prefix, as shown in Figure 7.22. This time, we cannot find one ISP that is superior in both metrics. Sprint ranks high in terms of reachability but ranks mediocre for stretch. In contrast, Cogent and Verio rank low under unreachability score but have small stretch. Some ISPs such as British Telecom ranks low in both metrics. In this case, customer may consider using Cogent as the primary ISPs and Sprint as the backup ISP to achieve their goal, e.g.shifting traffic from Cogent to Sprint when destination becomes unavailable. Note that in

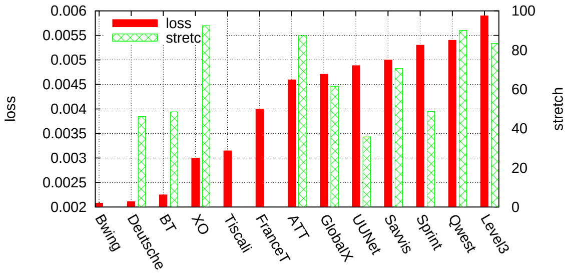


Figure 7.21: Correlation between loss rate and stretch between New York and Seattle.

practice these decisions likely to be made based on longer-term observations than the one month data used in this study.

7.5.3.3 Correlation between reachability and diversity

It is known that different reachability behavior across ISPs is likely caused by difference in their network topologies. Here we attempt to explain the observed reachability difference by correlating reachability with diversity. We measure diversity as the number of distinct AS-level paths from a particular PoP to one destination. Figure 7.23 shows the reachability to prefix 202.88.241.0/24 (an ISP in India) as well as the number of distinct paths towards it. For the ease of illustration, we use maximum diversity minus the actual diversity as y-axis on the right. Thus, smaller values indicate higher diversity. We observe relatively good correlation between reachability and diversity as expected, given that diverse paths can help bypass certain failures.

7.6 System Performance

We discuss the resource consumption of our system in terms of bandwidth, memory, and CPU usage to demonstrate its low overhead and thus feasibility as a light-

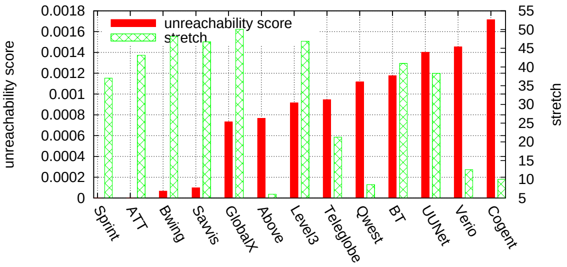


Figure 7.22: Correlation between unreachability and stretch to destination 202.65.134.0/24 (a company in HK).

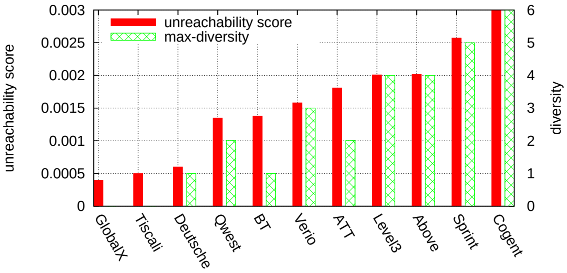


Figure 7.23: Correlation between unreachability and diversity to destination 202.88.241.0/24 (an ISP in India).

weight standalone distributed tool for comparing ISPs in real time using several metrics.

NetCmp was set up on the PlanetLab testbed using 186 probers on average, as restricted by the availability of usable probe nodes. Using the tasks assigned by the multi-ISP path selector the number of destinations each prober measures ranges from 4,058 to 12,000 destinations. The bandwidth consumption ranges from 17Kbps to 443Kbps for large packet experiments across all the probers for both reachability and loss rate measurements. Considering the trade-off between the load and the accuracy of detecting unreachability events, we use a redundancy factor of 3. The maximum memory consumption is 3.5GB memory in our prototype. The complexity of the optimization algorithm is $O(ep)$ (e: number of elements, p: number of src-dst pairs). We expect e to grow slowly with number of ISPs, as each src-dst pair traverses only a subset of ISPs. In the worst case, $e = p \times N$, where N is the number of ISPs. In the current prototype, the execution time of each run of the optimization component is around 25 minutes, which is only 41% of the entire probing interval, and can thus keep up with our experiments. Thus, we can conclude that NetCmp is quite scalable.

7.7 Conclusion

In this chapter we presented the design and implementation of a deployed system called NetCmp that accurately and scalably measure the performance of multiple ISP networks under several key performance metrics, covering latency, loss rate, path diversity, and destination reachability. Our system runs solely at end-hosts without requiring any proprietary internal ISP data or ISP cooperation. Using collaborative probes launched from many PlanetLab hosts for one month, we reveal the detailed differences in various network performance metrics across multiple ISPs.

The key results are summarized as follows. We found the network performance measured differs significantly across ISPs. The relative ranking of each ISP under different metrics can vary to a large degree as well. For instance, some ISPs with small internal latency can have many more unreachability events. The unreachability events are rare: the average availability of the destination paths are over 99.99% for all ISPs. But the relative difference between ISPs is significant. For example, the score of Teleglobe is over an order of magnitude larger than that of Bwing. By measuring multiple ISPs simultaneously, NetCmp has the unique opportunity to correlate ISPs at any time instance to analyze shared risks across ISPs. We witnessed that several ISPs can be highly correlated when the failures occurring close to the destination network. Finally, although the ranking is stable in short term period, it can change across several months, suggesting the necessity for continuous ISP monitoring, which our system is designed for.

Overall, our results show that ISPs' performance differs significantly across multiple dimensions, underscoring the importance of comprehensive evaluation of ISP performance. Our work provides valuable information for both customers and ISPs to make informed decisions about routing and peering selection. It also enables end-users to independently verify SLAs offered by their providers. In the long term, this provides incentives for ISPs to improve their network performance as such ISP performance data become widely available. We believe NetCmp is an important step to attaining better accountability and fairness on today's Internet.

CHAPTER VIII

Conclusion

8.1 Thesis summary and discussion

The objectives of this thesis is to enhance the reliability and performance of the Internet by designing effective monitoring and diagnosis systems to enable efficient recovery and prevention. To achieve this task, I take the approach of building large-scale, accurate and efficient network monitoring systems from purely end-systems perspective, which enables end hosts to diagnose and react to performance degradations in real-time. I have built a large-scale monitoring system running on 300 machines distributively which demonstrates the feasibility and value of my research in reality.

8.1.1 On the impact of route monitor selection: better monitor placement

This dissertation starts with systematically examining the visibility constraints imposed by the deployment of route monitors on answering diverse important research questions including Internet topology discovery, dynamic routing behavior detection, and inference of important network properties. Our study of route monitor selection demonstrates one fundamental limitation of ISP-centric approach on the routing plane: the results heavily depend on the available data set that the ISPs are willing to reveal.

The monitor placement strategies can differ significantly according to different objectives. Even with a single objective, it is generally NP-hard. For instance, under the context of detecting routing attacks, the goal of placing route monitors is to maximize the likelyhood that BGP attacks can be observed and hence detected.

8.1.2 Diagnosing routing disruptions: coverage and accuracy trade-offs

Motivated by the above observations, we develop two systems to diagnose two types of severe network disruptions: routing-induced disruptions and ISP policy induced traffic performance difference. We first presented the first system that accurately and scalably diagnoses routing disruptions purely from end-systems without access to any sensitive data such as BGP feed or router configurations from ISP networks. Using a simple greedy algorithm on two bipartite graphs representing observed routing events, possible causes, and the constraints between them, our system effectively infers the most likely causes for routing events detected through light-weight traceroute probes. We comprehensively validate the accuracy of our results by comparing with existing ISP-centric method, publicly-available router configurations, and network operators' mailing list.

The number of the probing location and the coverage of their network locations is a big concern in our approach. Limited location may make us not be able to detect certain routing change. Moreover, in the greedy algorithm, some real root causes with fewer number of support may not be selected due to the limited coverage. We are well aware of the coverage limitations of our system. To overcome that, we apply the common intuition that there are usually few concurrent causes explaining concurrent routing events to identify the most likely causes associated with the given AS. A straightforward solution to improving coverage is to use more end systems. We use all the available PlanetLab sites (roughly 200) to probe five target ISPs. As future work, we plan to study to how the coverage will improve our inference accuracy.

Though in reality the probing resource is limited, there are still a number of advantages of this approach. First, big network disruptions may be visible even with limited visibility. Therefore, covering a reasonable fraction of one target AS is sufficient to detect big disruptions. Second, this work is the first step to illustrate the feasibility of using the end system based approach for routing diagnosis. As the probing resource increases, this approach will become more and more powerful. Third, though from a given ISP's perspective, we cannot have full visibility, it is possible for us to monitor multiple ISPs. While sacrificing the full visibility from one particular

ISP, we gain the ability to monitor multiple ISPs.

Given limited CPU and network resources at end systems, we cannot probe every PoP-prefix pair as frequently as desired. This implies we may miss some routing events that occur between two consecutive probes. Our system focuses on diagnosing routing changes that are long-lived enough to warrant ISP's corrective action rather than those that are transient and repaired by itself quickly. In fact, we should avoid overwhelming ISPs by reporting too many transient changes. Given that inter-domain routing changes can require up to several minutes to converge, our system is sufficient to detect the significant changes. we believe our current probing frequency is sufficient to detect those long-lived and significant changes.

8.1.3 Traffic differentiation: detection and prevention

To detect general traffic differentiation, we presented the design and implementation of the first deployed system, NVLens , to accurately and scalably detect network neutrality violations performed by backbone ISP networks. Using collaborative probing from end hosts with various innovative application-specific probing in carefully designed task schedules, we demonstrate the surprising evidence of traffic discrimination carried out by today's backbone ISPs.

8.1.4 Applications of end-host based monitoring systems

Finally, we demonstrate the potentials for both short-term and long-term mitigation. To prevent being affected by abnormal routing changes, we develop an efficient framework to measure and predict data plane performance degradation as a result of routing changes. Using this framework, we conducted a large scale Internet measurement study and characterized data plane performance upon receiving a BGP routing update. We observe that the data plane performance of a certain set of prefixes is highly predictable. We further develop a statistical model which can accurately predict the severity of potential data plane failures based on observations of routing updates for a given prefix. We show that our model is very useful in a number of applications such as route selection in an overlay network.

In this section, we discuss potential applications of the measurement framework and the prediction model. First, the measurement infrastructure provides a platform of measuring the impact of routing changes on data plane performance. Detecting control plane changes and predicting corresponding data plane disruptions provide additional information for best route selection. Thus, it helps selecting best routes in terms of better data plane performance given alternate routes to the same destination.

By examining the predicted data plane performance among all available routes, the least impacted route can be selected as the best route to reduce the likelihood and degree of data plane performance degradation. we can select the routes which have least possibility of causing reachability problem.

Let us use an example to illustrate how the route selection process can be improved for overlay routing. Suppose a given destination prefix can be reached via multiple overlay nodes. When a failure is predicted in AS A based on the observation of a routing update of the destination prefix, we can select the next hop for a path avoiding A to reach the destination. Compared to random next hop selection, this selection process is more deterministic and has a higher chance of avoiding data plane failures.

In the long term, we presented the design and implementation of a deployed system called NetCmp that accurately and scalably measure the performance of multiple ISP networks under several key performance metrics, covering latency, loss rate, path diversity, and destination reachability. Our system runs solely at end-hosts without requiring any proprietary internal ISP data or ISP cooperation. Using collaborative probes launched from many PlanetLab hosts for one month, we reveal the detailed differences in various network performance metrics across multiple ISPs. Our results show that ISPs' performance differs significantly across multiple dimensions, underscoring the importance of comprehensive evaluation of ISP performance. Our work provides valuable information for both customers and ISPs to make informed decisions about routing and peering selection.

This proposed work is the first attempt to monitoring stability and SLA compliance of multiple ISPs simultaneously from the end-host users' perspective with low probing overhead. It is also the first system that uses end-host based measurement to detect traffic discrimination to ensure various applications' end-to-end performance. We believe our work is an important step to empowering customers and ISPs to attain better accountability on today's Internet.

8.2 Future work

In the course of my research, I have learned three general guidelines in network monitoring and troubleshooting. First, monitoring complex networks requires careful designs to balance accuracy, coverage and overhead. Second, network measurement data is often coarse-grained containing a large amount of noise and unavoidable artifacts. Systematic statistical techniques are needed to find the needle in the haystack. Third, in network troubleshooting, a single network event can often lead to different consequences at different network layer and in different locations. Thus, intelligent correlation across protocol layers and across geographic locations is greatly useful for uncovering the underlaying causes. I believe these three guidelines can be held generally in other types of networks.

In the near future, I am interested in identifying the new challenges of network monitoring in other types of networks, particularly large-scale Enterprise networks, data centers, and online social networks. The monitoring techniques in the Internet cannot be trivially applied. There are a number of interesting questions for research in this area.

8.2.1 Monitoring in Enterprise network and data center environment

Monitoring in Enterprise network and data center environment has more complex common underlying shared components at different levels. Todays data centers contain tens of thousands of computers running various applications ranging from static content-based web services to complex cloud computing. There are three key challenges in monitoring such networks. First, complex sharing structure exists. Machines in the same rack share common power failure. Clusters of machines connecting to same switches/routers share common network failure and bandwidth limits. Processes of same application running on different machines may share common logical failures, e.g., application level deadlock. Second, the requirements vary for monitoring at different network levels and for different applications. The trade-off mentioned in the first guideline should be considered separately. Third, the deployment is difficult in real production system. Unlike Internet, active monitoring in the data center environment is usually difficult given the cost, reliability, privacy, and security concerns. Designing monitoring systems with little overhead and without information leakage can help overcome the deployment obstacles.

8.2.2 Monitoring in online social network

Monitoring in online social network has larger challenges as the normal behavior is not well-defined. Learning techniques can be used to detect anomalous behavior. Social network properties such as degree distribution, connectivity, and communication patterns can be explored in constructing the monitoring system. Overall, I believe my previous work provides a solid background to address these challenging problems.

8.2.3 Analyzing the economical and technological factors in the Internet

My research so far has focused on enabling end users to gain sufficient information about the ISP networks instead of treating network as a black box. The topic of network neutrality is currently highly contentious for today's Internet. Previously, ISP networks are assumed to be neutral by carrying traffic without any preferential treatment. I am interested in exploring both economic factors and technological factors together with the monitoring results to reason about the necessity and benefit of various “net-neutrality” violation actions. For example, end users and the ISPs can be modeled as multiple parties on the market with equivalent information given the end-host based monitoring. We can study the profit gain, technology cost, and potential risk and panalty in having traffic differentiation. I would also be interested in new network architecture and protocol designs that could create a more open market model for the Internet.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Arbor Ellacoya e100 . http://www.arbornetworks.com.

[2] AT&T Continues to Adjust TOS to Limit 3G Video. http://www.nytimes.com/external/gigaom/2009/04/29/29gigaom-att-continues-to-adjust-tos-to-limit-3g-video-12208.html.

[3] AT&T Managed Internet Service (MIS). http://new.serviceguide.att.com/mis.htm.

[4] Geolocation and Online Fraud Prevention from MaxMind. www.maxmind.com.

[5] GNU Zebra-routing software. http://www.zebra.org.

[6] Internet2 Network NOC - Research Data. http://www.abilene.iu.edu/i2network/research-data.html.

[7] MPLS VPN Service Level Agreements. www.sprint.com/business/resources/mpls_vpn.pdf.

[8] NANOG Mailing List Information. http://www.nanog.org/mailinglist.html.

[9] NTT Communications Global IP Network Service Level Agreements (SLA). http://www.us.ntt.net/support/sla/network.

[10] Ripe NCC. http://www.ripe.net/ripencc/pub-services/np/ris/.

[11] Schooner User-Configurable Lab Environment. http://www.schooner.wail.wisc.edu/index.php3?stayhome=1.

[12] Sprint NEXTEL Service Level Agreements. http://www.sprint.com/business/support/serviceLevelAgreements.html.

[13] University of Oregon Route Views Archive Project. http://www.routeview.org.

[14] University of Oregon Route Views Archive Project. www.routeviews.org.

[15] S. Agarwal, C. Chuah, S. Bhattacharyya, and C. Diot. Impact of BGP Dynamics on Intra-Domain Traffic. In *Proc. ACM SIGMETRICS*, 2004.

[16] W. Aiello, J. Ioannidis, and P. McDaniel. Origin authentication in interdomain routing. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 165–178, New York, NY, USA, 2003. ACM.

[17] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. Symposium on Operating Systems Principles*, 2001.

[18] B. Augustin, M. Curie, T. Friedman, and R. Teixeira. Measuring Load-balanced Paths in the Internet. In *Proc. ACM SIGCOMM IMC*, 2007.

[19] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding Traceroute Anomalies with Paris Traceroute. In *Proc. of IMC*, 2006.

[20] I. Avramopoulos and J. Rexford. Stealth probing: Efficient data-plane security for IP routing. In *Proc. USENIX Annual Technical Conference*, 2006.

[21] I. Avramopoulos and J. Rexford. Stealth probing: Efficient data-plane security for IP routing. In *Proceedings of USENIX Annual Technical Conference*, 2006.

[22] I. Avramopoulos, J. Rexford, D. Syrivelis, and S. Lalis. Counteracting discrimination against network traffic. Technical Report TR-794-07, Princeton University Computer Science, 2007.

[23] P. Barford, A. Bestavros, J. Byers, and M. Crovella. On the marginal utility of network topology measurements. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 2001.

[24] G. Battista, M. Patrignani, and M. Pizzonia. Computing the Types of the Relationships Between Autonomous Systems. In *Proc. IEEE INFOCOM*, March 2003.

[25] G. Battista, M. Patrignani, and M. Pizzonia. Computing the Types of the Relationships Between Autonomous Systems. In *Proc. IEEE INFOCOM*, March 2003.

[26] L. Z. Beichuan Zhang, Daniel Massey. Destination Reachability and BGP Con-

vergence Time. In *Proc. of IEEE Globecom, Global Internet and Next Generation Networks*, 2004.

[27] R. Beverly, S. Bauer, and A. Berger. The Internet's Not a Big Truck: Toward Quantifying Network Neutrality. In *Proceedings of the 8th Passive and Active Measurement (PAM 2007) Conference*, 2007.

[28] V. J. Bono. 7007 Explanation and Apology. NANOG 97-04.

[29] A. Bremler-Barr, E. Cohen, H. Kaplan, and Y. Mansour. Predicting and Bypassing End-to-End Internet Service Degradations. In *Proc. ACM SIGCOMM Internet Measurement Conference*, 2002.

[30] T. Bu, N. Duffield, F. L. Presti, and D. Towsley. Network tomography on general topologies. *SIGMETRICS Perform. Eval. Rev.*, 30(1), 2002.

[31] R. Bush and T. Griffin. Integrity for virtual private routed networks. In *Proc. IEEE INFOCOM*, 2003.

[32] K. Butler, T. Farley, P. McDaniel, and J. Rexfod. A Survey of BGP Security Issues and Solutions. Technical Report Technical Report TD-5UGJ33, AT&T Labs - Research, 2004.

[33] K. Butler, P. McDaniel, and W. Aiello. Optimizing BGP security by exploiting path stability. In *Proc. CCS*, New York, NY, USA, 2006. ACM.

[34] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe. Design and Implementation of a Routing Control Platform. In *Proc.*

2nd Symposium on Networked Systems Design and Implementation (NSDI), 2005.

[35] H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Towards capturing representative AS-level Internet topologies. *Computer Networks*, 2004.

[36] Cisco Systems. Configuring Port to Application Mapping. `http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7c8.html`.

[37] Cisco Systems. Configuring Priority Queueing. `http://www.cisco.com/en/US/docs/ios/12_0/qos/configuration/guide/qcpq.html`.

[38] Cisco Systems. Simple network management protocol.

[39] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical Internet Coordinates for Distance Estimation. In *Proceedings of IEEE ICDCS*, March 2004.

[40] cPacket Networks Inc. Complete Packet Inspection on a Chip. `http://www.cpacket.com/`.

[41] J. Crowcroft. Net neutrality: the technical side of the debate: a white paper. *ACM Computer Communication Review*, 2007.

[42] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *Proceedings of ACM SIGCOMM*, August 2004.

[43] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy,

and G. Riley. AS Relationships: Inference and Validation. *ACM Computer Communication Review*, 37(1), 2007.

[44] X. Dimitropoulos and G. Riley. Modeling Autonomous System Relationships. In *Proc. of PADS*, 2006.

[45] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi. Detecting BitTorrent Blocking. In *Proc. ACM SIGCOMM Internet Measurement Conference*, 2008.

[46] Deep packet inspection. `www.networkworld.com/details/6299.html`.

[47] N. Duffield. Simple network performance tomography. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 210–215, New York, NY, USA, 2003. ACM.

[48] N. G. Duffield. Network tomography of binary network performance characteristics. *IEEE Transactions on Information Theory*, 52:5373–5388, 2006.

[49] J. P. Egan. *Signal Detection Theory and ROC Analysis.* Academic Press, New York, 1975.

[50] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251–262, New York, NY, USA, 1999. ACM.

[51] N. Feamster, D. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring

the effects of internet path faults on reactive routing. In *Proc. ACM SIGMETRICS*, 2003.

[52] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the Effects of Internet Path Faults on Reactive Routing. In *Proc. ACM SIGMETRICS*, Jun 2003.

[53] N. Feamster and H. Balakrishnan. Detecting BGP Configuration Faults with Static Analysis. In *Proc. Symposium on Networked Systems Design and Implementation*, May 2005.

[54] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs. Locating Internet Routing Instabilities. In *Proceedings of ACM SIGCOMM*, 2004.

[55] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. Gryniewicz, and Y. Jin. An Architecture for a Global Internet Host Distance Estimation Service. In *Proceedings of IEEE INFOCOM*, March 1999.

[56] L. Gao. On Inferring Autonomous System Relationships in the Internet. In *Proc. IEEE Global Internet Symposium*, 2000.

[57] G. Goodell, W. Aiello, T. Griffin, J. Ioanmidis, P. McDaniel, and A. Rubin. Working around BGP: an Incremental Approach to Improving Security and Accuracy in Interdomain Routing. In *Proc. NDSS*, 2003.

[58] R. Govindan and H. Tangmunarunkit. Heuristics for Internet Map Discovery. In *Proc. of IEEE INFOCOM*, pages 1371–1380, Tel Aviv, Israel, March 2000.

[59] Y. He, G. Siganos, M. Faloutsos, and S. V. Krishnamurthy. A systematic framework for unearthing the missing links: Measurements and Impact. In *Proc. of NSDI*, 2007.

[60] U. Hengartner, S. Moon, R. Mortier, and C. Diot. Detection and analysis of routing loops in packet traces. In *Proc. ACM SIGCOMM IMW*, 2002.

[61] X. Hu and Z. M. Mao. Accurate Real-time Identification of IP Prefix Hijacking. In *Proc. of IEEE Security and Privacy*, 2007.

[62] Y.-C. Hu, A. Perrig, and M. Sirbu. SPV: A Secure Path Vector Scheme for Securing BGP. In *Proc. ACM SIGCOMM*, 2004.

[63] Y. Huang, N. Feamster, A. Lakhina, and J. J. Xu. Diagnosing network disruptions with network-wide analysis. In *Proc. ACM SIGMETRICS*, pages 61–72, 2007.

[64] J. Karlin, S. Forrest, and J. Rexford. Pretty Good BGP: Improving BGP by cautiously adopting routes. In *Proc. International Conference on Network Protocols*, 2006.

[65] J. Karlin, J. Karlin, S. Forrest, and J. Rexford. Pretty Good BGP: Improving BGP by Cautiously Adopting Routes. In *Proc. of ICNP*, 2006.

[66] E. Karpilovsky and J. Rexford. Using forgetful routing to control BGP table size. In *Proc. CoNext*, 2006.

[67] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying Black Holes in the Internet with Hubble. In *Proc. of NSDI*, 2008.

[68] V. S. Kaulgud. IP Quality of Service: Theory and best practices. `www.sanog.org/resources/sanog4-kaulgud-qos-tutorial.pdf`, 2004.

[69] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE J. Selected Areas in Communications*, 2000.

[70] Keynote. Internet Health Report. `http://internetpulse.keynote.com/`.

[71] S. G. Kolliopoulos and N. E. Young. Approximation algorithms for covering/packing integer programs. *Journal of Computer and System Sciences*, 71(4), 2005.

[72] R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. IP fault localization via risk modeling. In *Proc. Symposium on Networked Systems Design and Implementation*, 2005.

[73] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proc. ACM SIGCOMM*, 2000.

[74] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *Proc. ACM SIGCOMM*, 2000.

[75] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. PHAS: A Prefix Hijack Alert System. In *Proc. of USENIX Security Symposium*, 2006.

[76] M. Lad, R. Oliveira, B. Zhang, and L. Zhang. Understanding resiliency of internet topology against prefix hijack attacks. In *Proc. of DSN*, 2007.

[77] J. Li, R. Bush, Z. M. Mao, T. Griffin, M. Roughan, D. Stutzbach, and E. Purpus. Watching Data Streams Toward a Multi-Homed Sink Under Routing Changes Introduced by a BGP Beacon. In *Proc. Passive and Active Measurement Workshop*, 2006.

[78] G. Lu, Y. Chen, S. Birrer, F. E. Bustamante, C. Y. Cheung, and X. Li. End-to-end inference of router packet forwarding priority. In *Proc. IEEE INFOCOM*, 2007.

[79] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An Information Plane for Distributed Services. In *Proc. Operating Systems Design and Implementation*, 2006.

[80] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring Link Weights Using End-to-End Measurements. In *Proc. ACM SIGCOMM IMW*, 2002.

[81] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet Path Diagnosis. 2003.

[82] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfigurations. In *Proc. ACM SIGCOMM*, August 2002.

[83] R. Mahajan, M. Zhang, L. Poole, and V. Pai. Uncovering Performance Differences in Backbone ISPs with Netdiff. In *Proceeding of NSDI*, 2008.

[84] A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao. Towards Automated Performance Diagnosis in a Large IPTV Network. In *Proc. ACM SIGCOMM*, August 2009.

[85] Z. M. Mao, R. Bush, T. G. Griffin, and M. Roughan. BGP beacons. In *Proc. ACM SIGCOMM Internet Measurement Conference*, 2003.

[86] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang. On AS-Level Path Inference. In *Proc. ACM SIGMETRICS*, 2005.

[87] Z. M. Mao, J. Rexford, J. Wang, and R. Katz. Towards an Accurate AS-Level Traceroute Tool . In *Proc. ACM SIGCOMM*, 2003.

[88] A. W. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. *SIGMETRICS Perform. Eval. Rev.*, 33(1):50–60, 2005.

[89] W. Muhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-topology model that captures route diversity. In *Proc. ACM SIGCOMM*, 2006.

[90] W. Muhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-Topology Model. In *Proc. of ACM SIGCOMM*, 2006.

[91] S. Murphy. BGP Vulnerabilities Analysis. IETF draft June 2003.

[92] Neil Spring and Ratul Mahajan and David Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proc. ACM SIGCOMM*, 2002.

[93] J. Ng. Extensions to BGP to Support Secure Origin BGP (soBGP). IETF Draft: draft-ng-sobgp-bgp-extensions-01.txt, November 2002.

[94] T. S. E. Ng and H. Zhang. Predicting Internet Network Distance with Coordinates-Based Approaches. In *Proceedings of IEEE INFOCOM*, June 2002.

[95] R. Oliveira, M. Lad, B. Zhjang, D. Pei, D. Massey, and L. Zhang. Placing BGP Monitors in the Internet. Technical Report, UCLA CS Department, TR 060017, May 2006.

[96] R. V. Oliveira, B. Zhang, and L. Zhang. Observing the evolution of internet as topology. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 313–324, New York, NY, USA, 2007. ACM.

[97] A. Pathak, H. Pucha, Y. Zhang, Y. C. Hu, and Z. M. Mao. A Measurement Study of Internet Delay Asymmetry. In *Proc. Passive and Active Measurement Conference (PAM)*, 2008.

[98] V. Paxson. End-to-end routing behavior in the Internet. In *Proc. the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, volume 26,4 of *ACM SIGCOMM Computer Communication Review*. ACM Press, 1996.

[99] D. Pei, L. Wang, D. Massey, S. F. Wu, and LixiaZhang. A Study of Packet Delivery Performance during Routing Convergence. In *Proc. of IEEE International Conference on Dependable Systems and Networks (DSN)*, 2003.

[100] D. Pei, X. Zhao, D. Massey, and L. Zhang. A Study of BGP Path Vector Route Looping Behavior. In *Proc. IEEE International Conference on Distributed Computing Systems*, 2004.

[101] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A Blueprint for Introducing Disruptive Technology Into the Internet. In *Proc. of ACM HotNets*, 2002.

[102] PlanetLab. `http://www.planet-lab.org`.

[103] H. Pucha, Y. Zhang, Z. M. Mao, and Y. C. Hu. Understanding network delay changes caused by routing events. In *Proc. ACM SIGMETRICS*, 2007.

[104] D. Raz and R. Cohen. The internet dark matter: on the missing links in the as connectivity map. In *Proc. IEEE INFOCOM*, 2006.

[105] Y. Rekhter and T. Li. A border gateway protocol. RFC 1771, 1995.

[106] Y. Shinoda, K. Ikai, and M. Itoh. Vulnerabilities of passive internet threat monitors. In *14th USENIX Security Symposium*, August 2005.

[107] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos. Power laws and the as-level internet topology. *IEEE/ACM Trans. Netw.*, 11(4):514–524, 2003.

[108] J. Sommers and P. Barford. An Active Measurement System for Shared Environments. In *Proc. ACM SIGCOMM IMC*, 2007.

[109] J. Sommers, P. Barford, N. Duffield, and A. Ron. Improving accuracy in end-to-end packet loss measurement. *SIGCOMM Comput. Commun. Rev.*, 35(4):157–168, 2005.

[110] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, 2004.

[111] N. Spring, D. Wetherall, and T. Anderson. Reverse-Engineering the Internet. In *Proc. First ACM SIGCOMM HotNets Workshop*, 2002.

[112] N. Spring, D. Wetherall, and T. Anderson. Scriptroute: A Public Internet Measurement Facility. In *Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS)*, 2003.

[113] A. Sridharan, Sue.B.Moon, and C. Diot. On the Correlation between Route Dynamics and Routing. In *Proc. ACM SIGCOMM IMC*, October 2003.

[114] A. Stuart, K. Ord, and S. Arnold. *Kendall's Advanced Theory of Statistics.* London: Arnold, 1999.

[115] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proc. IEEE INFOCOM*, 2002.

[116] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz. Listen and Whisper: Security Mechanisms for BGP. In *Proc. first Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

[117] H. Tangmunarunkit, J. Doyle, R. Govindan, W. Willinger, S. Jamin, and S. Shenker. Does as size determine degree in as topology? *SIGCOMM Comput. Commun. Rev.*, 31(5):7–8, 2001.

[118] M. B. Tariq, M. Motiwala, and N. Feamster. NANO: Network Access Neutrality Observatory . In *Proceedings of ACM HotNets-V, Calgary, Alberta, Canada*, 2008.

[119] R. Teixeira and J. Rexford. A measurement framework for pin-pointing routing changes. In *NetT '04: Proceedings of the ACM SIGCOMM workshop on Network troubleshooting*, pages 313–318, 2004.

[120] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in ip networks. *SIGMETRICS Perform. Eval. Rev.*, 32(1):307–319, 2004.

[121] J. Tukey. *Bias and confidence in not quite large samples.* Ann. Math. Statist, 1958.

[122] F. Wang, L. Gao, J. Wang, and J. Qiu. On Understanding of Transient Interdomain Routing Failures. In *Proc. International Conference on Network Protocols*, 2005.

[123] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush. A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. In *Proc. ACM SIGCOMM*, 2006.

[124] C. Wright, F. Monrose, and G. Masson. On inferring application protocol behaviors in encrypted network traffic. In *Journal of Machine Learning Research (JMLR): Special issue on Machine Learning for Computer Security*, 2006.

[125] J. Wu, Z. M. Mao, J. Rexford, and J. Wang. Finding a needle in a haystack: Pinpointing significant BGP routing changes in an IP network. In *Proc. Symposium on Networked Systems Design and Implementation*, 2005.

[126] J. Wu, Y. Zhang, Z. M. Mao, and K. Shin. Internet Routing Resilience to Failures: Analysis and Implications. 2007.

[127] J. Xia, L. Gao, and T. Fei. Flooding Attacks by Exploiting Persistent Forwarding Loops. In *Proc. ACM SIGCOMM IMC*, 2005.

[128] X. Yang. Auction, but Don't Block. Work in Progress.

[129] X. Yang, G. Tsudik, and X. Liu. A Technical Approach to Net Neutrality. In *Proceedings of ACM HotNets-V, Irvine*, 2006.

[130] H. Yin, B. Sheng, H. Wang, and J. Pan. Securing BGP through keychain-based signatures. In *Proceedings of the 15th IWQoS'07*, June 2007.

[131] A. Zeitoun and S. Jamin. Rapid Exploration of Internet Live Address Space Using Optimal Discovery Path. In *Proc. Global Communications Conference*, 2003.

[132] B. Zhang, V. Kambhampati, M. Lad, D. Massey, and L. Zhang. Identifying BGP Routing Table Transfers. In *Proc. SIGCOMM Mining the Network Data (MineNet) Workshop*, August 2005.

[133] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the internet as-level topology. *SIGCOMM Comput. Commun. Rev.*, 35(1):53–61, 2005.

[134] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level Topology. *ACM SIGCOMM Computer Communication Review, special issue on Internet Vital Statistics*, 2005.

[135] M. Zhang, Y. Ruan, V. Pai, and J. Rexford. How DNS Misnaming Distorts Internet Topology Mapping. In *Proceedings of USENIX Annual Technical Conference*, 2006.

[136] M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang. PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services. In *Proc. Symposium on Operating Systems Design and Implementation*, 2004.

[137] Y. Zhang, M. Mao, and M. Zhang. Ascertaining the Reality of Network Neutrality Violation in Backbone ISPs. In *Proceedings of ACM HotNets-V, Calgary, Alberta, Canada*, 2008.

[138] Y. Zhang, Z. M. Mao, and J. Wang. A Framework for Measuring and Predicting the Impact of Routing Changes. In *Proc. IEEE INFOCOM*, 2007.

[139] Y. Zhang, Z. M. Mao, and M. Zhang. Effective Diagnosis of Routing Disruptions from End Systems. In *Proceedings of NSDI*, 2008.

[140] Y. Zhang, Z. Zhang, Z. M. Mao, and Y. C. Hu. On the Impact of Route Monitor Selection. In *Proc. ACM SIGCOMM IMC*, 2007.

[141] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An Analysis of BGP Multiple Origin AS (MOAS) Conflicts. In *Proc. ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[142] X. Zhao, B. Zhang, A. Terzis, D. Massey, and L. Zhang. The Impact of Link Failure Location on Routing Dynamics: A Formal Analysis. In *Proc. of ACM SIGCOMM Asia Workshop*, 2005.

[143] Z. Zhong, R. Keralapura, S. Nelakuditi, Y. Yu, J. Wang, C.-N. Chuah, and S. Lee. Avoiding Transient Loops through Interface-Specific Forwarding. In *Proc. IFIP/IEEE IWQoS*, June 2005.

[144] T. Zseby. Deployment of sampling methods for sla validation with non-intrusive measurements. In *Proc. of Passive and Active Measurement workshop*, 2001.