

Electronic Theses and Dissertations, 2004-2019

2013

Measuring The Evolving Internet Ecosystem With Exchange Points

Mohammad Zubair Ahmad
University of Central Florida

 Part of the [Computer Engineering Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd>
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Ahmad, Mohammad Zubair, "Measuring The Evolving Internet Ecosystem With Exchange Points" (2013).
Electronic Theses and Dissertations, 2004-2019. 2595.
<https://stars.library.ucf.edu/etd/2595>

MEASURING THE EVOLVING INTERNET ECOSYSTEM WITH EXCHANGE
POINTS

by

MOHAMMAD ZUBAIR AHMAD

B.E. Information Science and Engineering, Visvesvariah Technological University, 2005

M.S. Computer Engineering, University of Central Florida, 2007

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2013

Major Professor: Ratan Guha

© 2013 Mohammad Zubair Ahmad

ABSTRACT

The Internet ecosystem comprising of thousands of Autonomous Systems (ASes) now include Internet eXchange Points (IXPs) as another critical component in the infrastructure. Peering plays a significant part in driving the economic growth of ASes and is contributing to a variety of structural changes in the Internet. IXPs are a primary component of this peering ecosystem and are playing an increasing role not only in the topology evolution of the Internet but also inter-domain path routing. In this dissertation we study and analyze the overall affects of peering and IXP infrastructure on the Internet. We observe IXP peering is enabling a quicker flattening of the Internet topology and leading to over-utilization of popular inter-AS links. Indiscriminate peering at these locations is leading to higher end-to-end path latencies for ASes peering at an exchange point, an effect magnified at the most popular worldwide IXPs. We first study the effects of recently discovered IXP links on the inter-AS routes using graph based approaches and find that it points towards the changing and flattening landscape in the evolution of the Internet's topology. We then study more IXP effects by using measurements to investigate the networks benefits of peering. We propose and implement a measurement framework which identifies default paths through IXPs and compares them with alternate paths isolating the IXP hop. Our system is running and recording default and alternate path latencies and made publicly available. We model the probability of an alternate path performing better than a default path through an IXP

by identifying the underlying factors influencing the end-to-end path latency. Our first-of-its-kind modeling study, which uses a combination of statistical and machine learning approaches, shows that path latencies depend on the popularity of the particular IXP, the size of the provider ASes of the networks peering at common locations and the relative position of the IXP hop along the path. An in-depth comparison of end-to-end path latencies reveal a significant percentage of alternate paths outperforming the default route through an IXP. This characteristic of higher path latencies is magnified in the popular continental exchanges as measured by us in a case study looking at the largest regional IXPs.

We continue by studying another effect of peering which has numerous applications in overlay routing, Triangle Inequality Violations (TIVs). These TIVs in the Internet delay space are created due to peering and we compare their essential characteristics with overlay paths such as detour routes. They are identified and analyzed from existing measurement datasets but on a scale not carried out earlier. This implementation exhibits the effectiveness of GPUs in analyzing big data sets while the TIVs studied show that a set of common inter-AS links create these TIVs. This result provides a new insight about the development of TIVs by analyzing a very large data set using GPGPUs.

Overall our work presents numerous insights into the inner workings of the Internet's peering ecosystem. Our measurements show the effects of exchange points on the evolving Internet and exhibits their importance to Internet routing.

Oh dear dad

Can you see me now?

I am myself

Like you somehow.

ACKNOWLEDGMENTS

Throughout the course of the my time here at UCF I have had the opportunity to meet, interact and work with wonderful people, both faculty and students. My advisor has been the first and greatest influence not only in determining my research directions but also other critical aspects of life in general. He provided me the necessary freedom to think and work freely while putting in perspective all the outlandish ideas I would come up with during the course of our numerous conversations. His guidance and attention to detail throughout the PhD process has been invaluable and it has taught me how to first define a problem and then the basic approach towards trying to solve it. Along with the research aspects, I also learnt life lessons in patience, persistence and creating the drive to work towards my goal. Many thanks once again to Dr. Guha for being an almost perfect advisor.

I have had the chance to work with a few other faculty over the course of my research and I would take this opportunity to thank other members of my dissertation committee. Dr. Bassiouni for his brief but extremely insightful observations and Dr. Goldiez for actually hiring me as a research assistant for the various external projects we worked on at the IST along with his unique view of looking at the big picture while we would get lost in the details. Dr. Jha helped me immensely in the final stages of my dissertation to get the modeling studies off the ground and timely insights when I would hit the proverbial wall. And lastly,

Dr. Chatterjee who would always spend a lot of time with me discussing everything from the smaller details to the general roadmap of my dissertation.

Multiple years in the program equates to a lot of hours in the lab. And this is where friends and lab mates turn out to be critical cogs in the PhD wheel. All the hours spent in the lab (and outside at the coffee shop) with Ilhan, discussing research, basketball and heavy metal (along with all the shows) will be sorely missed. The numerous daily coffee breaks with Saptarshi, Swastik and Mukund talking about movies, soccer and tv shows was something I would look forward to almost every day. Writing civil engineering code for Melike and setting up her computer over most weekends was fun (at times) but the cookies, coffee and entertainment she bought to the lab has always been very much appreciated. Hanging out at the beach, weekly lunches and the yearly Dave Matthews Band road trips with Megan resulted in totally awesome times. Engaging discussions with Faraz about theoretical computer science or programming in general helped stimulate my thoughts towards the basics of computation, something I have always wanted to work more on. Outside school, many thanks go to Jorge for being a friend and a cool boss at my web development job while last but not the least Aldo and Jake for introducing me to mountain biking and being great friends. The saturday morning rides followed by gorging on food were perfect stress busters and the dozens of metal and rock shows we went to over the years have just been memorable. My heartfelt thanks to all of you.

In the end this would not have been possible without the pillars of support my family has been throughout the entire process. Sayeed bhai and Riya apu were always encouraging

(and feeding me when I would get bored of my own cooking) while my brother Junaid (and later) my sister-in-law Nazia kept providing the support even when I had grave doubts about myself. They had a greater level of belief in me than me. And finally my Mom for whom I do not need to say much except that this work has come to fruition only because of her. What kept me going was thinking about her and how much she wanted to see me succeed. This work is for her.

TABLE OF CONTENTS

LIST OF FIGURES	xviii
LIST OF TABLES	xxii
CHAPTER 1: INTRODUCTION	1
1.1 Motivation	1
1.2 Approach	3
1.3 Contribution	5
1.4 Organization	6
CHAPTER 2: INTERNET EXCHANGE POINTS	7
2.1 IXP architecture	7
2.2 IXP Growth	9
2.3 Data sources and identifying IXP peering links from traceroutes	11

2.4	Routing Performance	14
2.5	Internet Topology	15
2.6	Triangle Inequality Violations	16
CHAPTER 3: RELATED WORK		19
3.1	Routing Performance	19
3.2	Internet Topology	20
3.3	Triangle Inequality Violations	21
CHAPTER 4: INTERNET TOPOLOGY EVOLUTION		24
4.1	AS graph analysis	25
4.1.1	Graph Construction	26
4.1.2	Validity of chosen datasets	29
4.2	Topology characteristics	31
4.2.1	Degree distribution	31
4.2.2	Power law degree distributions	33

4.2.3	Joint degree distribution	35
4.2.4	Clustering coefficient	37
4.2.5	Rich club connectivity	40
4.2.6	Node coreness	42
4.2.7	Distance and eccentricity	43
4.2.8	Betweenness	45
4.3	Analysis and discussions	47
CHAPTER 5: BANDWIDTH MEASUREMENTS		52
5.1	Bandwidth studies of popular web destinations	52
5.1.1	Source and destinations	53
5.1.2	Route characteristics studied	54
5.1.3	Tools used	55
5.1.4	Filtering IXP route destinations from each probing source	56
5.2	Pathneck measurements and results	57

5.2.1	Identifying choke points at IXP locations	58
5.2.2	IXP route persistence	59
5.2.3	Link losses and queuing delays	62
5.2.4	End to end delay results	66
CHAPTER 6: TRIANGLE INEQUALITY VIOLATIONS DUE TO IXPs		68
6.1	Internet triangle inequality violations	68
6.2	Experiment setup	71
6.2.1	Dataset selection	71
6.2.2	Identifying end to end TIVs due to IXPs	72
6.2.3	Modules and parallel implementation	75
6.3	TIV Characteristics	76
6.3.1	Detour path graph properties	80
6.3.2	Graph characteristics	83
6.3.3	Degree distribution	84

6.3.4	Joint degree distribution	85
6.3.5	Average node coreness	86
6.3.6	Betweenness	87
6.4	Discussions and conclusions	89
CHAPTER 7: PARALLEL FRAMEWORKS FOR ANALYSIS WITH GPU		92
7.1	Platforms	94
7.2	Pattern matching in parallel	96
7.2.1	Parallel CPU implementation and timing results	97
7.2.2	GPU implementation details	98
7.2.3	CUDA implementation timing results	99
7.2.4	Diminished PFAC speedup?	101
7.3	All pairs shortest path in parallel	103
7.3.1	Implementation details	103
7.3.2	Timing results	104

7.3.3	Graph sizes	106
7.3.4	Recursive APSP on different AS graphs	110
7.4	Overall results	111
CHAPTER 8: IXP ROUTING PERFORMANCE		119
8.1	Path Selection	120
8.1.1	IXP path selection	120
8.1.2	Alternate paths: Common provider direct - Type 1	121
8.1.3	Alternate paths: Common provider indirect - Type 2	122
8.1.4	Detour Paths	123
8.1.5	Policy Compliance	124
8.2	Measurement Framework	125
8.2.1	Dataset selection	126
8.2.2	AS path generator	127
8.2.3	AS relationship identifier	128

8.2.4	Detour path generator	128
8.2.5	Alternate path generator	129
8.2.6	Path validator	129
8.2.7	Latency estimator	130
8.2.8	Path latency estimation	130
8.3	Overall evaluation	132
8.3.1	Dataset analyzed	132
8.3.2	Metrics	133
8.3.3	Comparisons with IXP paths	134
8.3.4	Comparisons with detour paths	135
8.3.5	Weekly comparisons	136
8.4	Evaluating popular IXPs	137
8.4.1	Dataset details	139
8.4.2	Selecting popular IXPs	140

8.4.3	Type 1 Paths: Common provider to destination direct	141
8.4.4	Type 2 Paths: Common provider indirect	144
8.4.5	Common provider characteristics	146
8.4.6	Evaluating provider latencies	147
8.5	Limitations	148
CHAPTER 9: IXP PATH MODELING		151
9.1	Background: GLM	151
9.2	Predictors for the GLM	152
9.3	Computing provider AS degree	153
9.4	Generating path data	155
9.5	Identifying best fit	155
9.6	Predictor variable effects	160
9.7	Discussion	162
CHAPTER 10: CONCLUSIONS		164

LIST OF REFERENCES 167

LIST OF FIGURES

Figure 1.1	Simple example of peering at IXP	4
Figure 2.1	A set of ASes transmitting data to each other through the Internet. . .	8
Figure 2.2	A set of ASes peering at an IXP	9
Figure 2.3	Percentage of IXP routes visible in one cycle of Skitter traceroute data every year for the month of September.	10
Figure 4.1	Node degree distribution. Power law behavior remain evident for all three datasets.	31
Figure 4.2	CCDF of node degree distribution for the three datasets. Power law behavior remains consistent across datasets.	32
Figure 4.3	Normalized average neighbor connections.	35
Figure 4.4	CCDF of average neighbor connections.	36
Figure 4.5	Local clustering with increasing node degrees.	38
Figure 4.6	CCDF of local clustering values.	39
Figure 4.7	CCDF of the Rich club connectivity (RCC) for the three graphs.	40
Figure 4.8	Average node coreness with increasing node degrees.	41
Figure 4.9	Distance distribution of three graphs.	43

Figure 4.10	Eccentricity distribution of three graphs.	44
Figure 4.11	Normalized node betweenness for three graphs.	45
Figure 4.12	CDF of log of edge betweenness for the three graphs.	45
Figure 4.13	Scatter plot showing edge betweenness centrality for three graphs.	46
Figure 5.1	Comparing choke point occurrences in IXPs.	57
Figure 5.2	CDF of route persistence values of IXP paths for both IP and AS level granularity.	61
Figure 5.3	Loss position of bottleneck point for IXP hops with respect to the bottleneck position.	62
Figure 5.4	CDF of distance from loss point to IXP bottleneck point.	64
Figure 5.5	CDF of queuing delays at bottleneck IXP links and non-bottleneck links.	65
Figure 5.6	Average end-to-end delay obtained from probing sources for IXP and non IXP destinations.	66
Figure 6.1	A simple example of a triangle inequality violation.	69
Figure 6.2	The sequential and parallel modules in the TIV identification process.	75
Figure 6.3	Observed TIV severity information.	76
Figure 6.4	Median delays and severities for TIVS generated due to peering.	78
Figure 6.5	Detour path characteristics. One intermediate hop show greater severities and larger latency savings.	79

Figure 6.6	Simplified example of high level and AS-level edges.	81
Figure 6.7	Node degree distributions of the High-level and AS-level graphs.	82
Figure 6.8	Average neighbor connections and node coreness.	85
Figure 6.9	Graph properties of AS level and High level graphs.	91
Figure 7.1	AS locations of computed TIVs in our study.	94
Figure 7.2	Pattern matching with PFAC in CUDA.	97
Figure 7.3	Timing comparison between serial and parallel processes for pattern matching in our traceroute data.	100
Figure 7.4	Pattern matching per prefix.	102
Figure 7.5	Speedups observed in APSP with increasing number of nodes.	108
Figure 8.1	The various paths measured in our framework between a set of end hosts.	124
Figure 8.2	System model depicting every component in the proposed framework infrastruc- ture.	125
Figure 8.3	CDF of RTT differences and RTT Difference ratios of alternate paths compared to IXP paths.	134
Figure 8.4	Weekly comparison of the percentage of better type 1 and type 2 alternate paths in comparison to the default paths.	136
Figure 8.5	CDF of RTT differences and RTT Difference ratios for paths from the common provider to the destination.	141

Figure 8.6	CDF to RTT difference ratios for type 1 paths compared to the best available detour path.	142
Figure 8.7	CDF of RTT differences and RTT Difference ratios for paths from the common provider to the destination through the second participant.	143
Figure 8.8	CDF of RTT differences and difference ratios for computed latencies from type 2 paths in comparison to the best available detour latencies.	144
Figure 8.9	CDF of number of common providers for participant ASes.	145
Figure 8.10	CDF of latency severities for type 1 paths.	146
Figure 9.1	Residual plots for the best fit model identified.	158
Figure 9.2	CDF plot of residuals.	159
Figure 9.3	Residual plots for the best fit model identified for number of participants.	159
Figure 9.4	Residual plots for the best fit model identified for average degree.	160
Figure 9.5	Residual plots for the best fit model identified for average hop locations.	160

LIST OF TABLES

Table 2.1	IXP growth obtained from searching known IXP prefixes from one cycle of Skitter data for the month of September	12
Table 4.1	Datasets analysed and nomenclature	25
Table 4.2	Comparing the number of observed links in the <i>PCH</i> , <i>RVIEWS</i> and <i>IXPMAP</i> graphs	26
Table 4.3	Table detailing graph summary statistics	50
Table 5.1	Some important PlanetLab probing source used and the number of IXP routes visible from each source to top 1000 websites.	56
Table 6.1	Comparing the number of observed links in the high level (G_h) and AS level graphs (G_a)	83
Table 7.1	Timing results (in secs) for the different serial iterative, recursive and different parallel implementations carried out in our experiments.	105
Table 7.2	Comparing APSP times for graphs of various sizes. Speedups obtained upto 8192 vertices remain consistent with previous work but with sizes greater than 8192, our clustering algorithm first runs to reduce the total number of vertices which in turn almost doubles the total APSP run times and reduces the speedups obtained.	109

Table 7.3 APSP with graph characteristics. Every CAIDA cycle generates equally large graphs with similar average edge weights resulting in very similar APSP completion times.

110

Table 7.4 Running times (in secs) for increasing number of nodes and routes. The number of routes measured is increased and the number of visible ASes in the routes denote the nodes. The effect of the serial modules of the TIV identification algorithm control overall times observed. 113

Table 7.5 Comparing total run times (in secs) for the entire process. Refer to table 7.6 for breakup of individual modules.Speedups for the proposed implementation using GPU's range from 3x-6x versus a serial implementation while in comparison to the best parallel implementation speedups range between 2-4x. 115

Table 7.6 Table detailing overall performance results. All times in seconds. The route search components takes a third less time while APSP decreases processing times from a day to less than a minute. 116

Table 8.2 The popular IXPs selected and their respective properties (from PCH [1] on 01/22/2012). T=TeraBytes, G=GigaBytes 138

Table 8.1 Paths analyzed from iPlane date ranges with number of IXP paths found along with the number of alternates generated.(M=Million) 150

Table 9.1 R-Squared values for all models with 5 fold cross-validation, x_1 = Number of IXP Participants, x_2 = Average common provider degree, x_3 = IXP hop location; $X_1 : x_2$ denotes an interaction between the respective variables. 157

CHAPTER 1: INTRODUCTION

Internet eXchange Points (IXPs) have recently grown into an integral component of the global Internet ecosystem. They facilitate the easy setup of peering between a wide variety of Autonomous Systems (ASes) in spite of their diverse business and technology policies. The dynamics of peering necessitates mutual agreement between a pair of ASes to share traffic based on a set of predefined criteria. Economic and network profits are the final goal. The direct exchange of traffic (instead of using a higher tier transit provider) enables the peering ASes to make significant savings in its transit costs to the larger provider(s) of which they are customers.

1.1 Motivation

Figure 1.1 exhibits an example of ASes setting up a peering relationship (AS A and B) at an IXP to exchange traffic while ASes E and F use traditional transit providers to direct their traffic.

The advent of peering has led to a gradual change in the fundamental routing structure of the Internet. Recent studies [2, 3] show the Internet to be evolving into a *flatter* system from the traditional hierarchical system. The flatter system of peering could be classified

into two broad types: *private*, where very large corporations set up their own dedicated (and expensive) backbone infrastructure; and *public*, where small, medium and large networks connect at an IXP located at a suitable geographic location. The peering exchange often times helps avoid long intra (or sometimes inter) continental backhaul transit links for traffic destined locally and thus not only helps save on transit costs but also improves network performance for ASes exchanging traffic at the exchange. Figure 1.1 exhibits an example of ASes setting up a peering relationship (AS A and B) at an IXP to exchange traffic while ASes E and F use traditional transit providers to direct their traffic.

The exchange ecosystem of the Internet affects the workings of the Internet in a variety of ways [4]. Peering affects the growth and evolution of the inter-domain AS topology which in turn significantly affects end-to-end routing. While peering has definite economic advantages for the lower tier-ASes, the network benefits of peering have not been studied by the research community. Networks (both service providers or other organizations) require greater knowledge in the determining the effectiveness of the peering points. This would result in informed and beneficial routing policies and ultimately more efficient packet routing across the Internet which coupled with the economic advantages of peering would lead to greater synergy between the networks.

1.2 Approach

In our work, we study the effects of IXPs and the peering fabric on the Internet architecture and routing dynamics. We study how peering is affecting the growth and evolution of the inter-domain AS topology and the effects on they are having on end-to-end routing. Some applications of peering are investigated in overlay routing along-with the design and implementation of a measurement framework. A recent work by Ager et. al. [5] identify more peering links at a single European IXP than the total number of these links known in 2010, a significant and rather startling finding. This incredibly rich peering fabric will have a definite impact on end-to-end path latencies, a characteristic which has not been actively studied by the research community. Our work is thus a first step in this direction where we actively analyze paths through the peering exchanges and determine their efficiency. This analysis of the constantly evolving peering fabric is carried out on a snapshot of the Internet's topology and data traffic. Here we look at how Internet path latencies are being affected by the phenomenon of worldwide peering. We compare end-to-end path latencies of paths through IXPs with synthetic alternate paths isolating the IXP effects. Such an approach helps pinpoint the effects IXPs are having on these paths. Using the measurements generated from our routing analysis framework we learn and identify a generalized linear regression model identifying the underlying factors affecting the latencies of the paths through the IXPs.

The increasing deployment of Internet eXchange Points (IXPs) has on the other hand led to a new avenue of research in determining additional links in the Internet topology. He et al. [6] present a framework which extracts new topology information from select IXPs and verify these edges from existing BGP tables and traceroute data. They report a higher number of new edges, most of them being of the peer-to-peer type. using available graph-theoretic methods and available data sources we study AS visibility at IXPs with the primary aim of establishing the role of these IXPs in determining the evolving Internet topology. We try to find out if IXP data presents significant connectivity information not present in the more conventional data sources such as RouteViews BGP data [7] or Skitter data from CAIDA [8] among others.

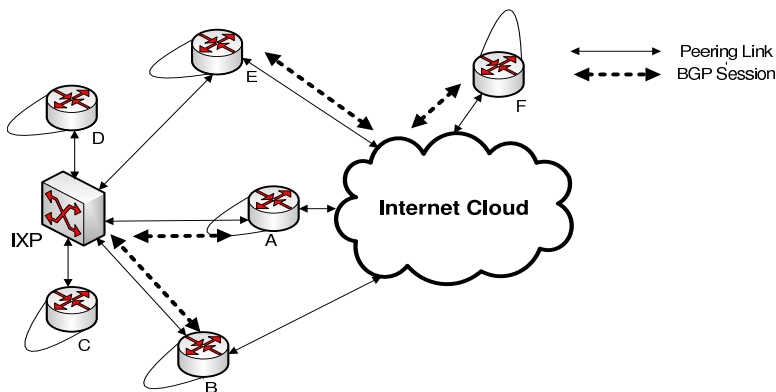


Figure 1.1 An example of peering at an IXP. AS A and B bypass the Internet to exchange traffic at the IXP switch. However ASes E and F need to set up BGP sessions to transmit data to each other through the Internet’s hierarchical routing infrastructure. A and B could also set up peering sessions easily with the other IXP participants such as C,D and E based on their peering policies.

1.3 Contribution

Our in-depth study and analysis of exchange points presents significant contributions towards a better understanding of the workings of the peering fabric of the Internet. We design and implement a measurement framework to infer path latencies of alternate paths isolating the IXP effects enables us to observe the high rate of over-utilization of default Internet paths through the public exchange points. We observe that one of ten IXP paths is the best available path amongst all other Internet paths, a characteristic indicating the potential of proper planning in the design and selection of an IXP for a peering relationship between participating ASes. Using the measurement framework we model the underlying dynamics of the IXP paths and observe that the number of IXP participants, size of provider ASes of peering networks and the relative location of an IXP along the path exhibit a direct influence on the performance of an alternate path. we observe IXP paths to be over-utilized in comparison with similar alternate paths between the same end hosts. One of ten IXP paths is the best available path amongst all other Internet paths, a characteristic indicating the potential of proper planning in the design and selection of an IXP for a peering relationship between participating ASes. Other characteristics of inter-domain Internet routing namely detours and the formation of triangle inequality violations (TIVs) largely remain consistent with that seen in previous studies, even with the advent of peering and a definite change in the Internet's topology evolution. We observe most IXP paths still possessing efficient detour alternatives due to the creation of TIVs in the Internet delay space.

Overall, this dissertation presents useful insight into the workings of route dynamics at the exchange points across the world. The switching networks at these locations are responsible for huge amounts of traffic everyday and play a major role in determining network services for millions of end-users. By pointing out the potential for improvements at these major locations, the lessons learnt here will be applicable to a large cross-section of ASes comprising of the peering fabric of the Internet.

1.4 Organization

This dissertation is organized as follows: chapter 2 presents relevant background literature about IXPs and its applications to Internet routing and topology and is followed by brief description of related work in chapter 3. Chapter 4 talks about the effects of peering links on Internet topology evolution. This is followed by chapter 5 which talks about the bandwidth studies to the popular web destinations and chapter 7 presenting details about parallel analysis techniques developed for network measurements. Measuring inter-domain routing performance is presented in chapter 8 which contains the proposed measurement framework and evaluation results and the modeling effort is detailed in chapter 9. We conclude in chapter 10 with discussions and the course for future work.

CHAPTER 2: INTERNET EXCHANGE POINTS

2.1 IXP architecture

IXPs are independently maintained physical infrastructures enabling public peering of member ASes. An IXP provides physical connectivity between the different member networks while the decision to initiate BGP sessions between AS pairs is left to the individual networks themselves. Figure 2.1 represents a regular scenario where a set of ASes (A to E) transmit data to each other using the Internet. Here local ASes end up using international links to transmit data which increases costs while decreasing network performance. Only if ASes have a local connection (AS C and D) are these problems mitigated. IXPs enable public peering between member ASes by providing physical connectivity infrastructure and the decision to initiate BGP sessions between AS pairs is left to the individual AS networks themselves. Most IXPs connect members through a common layer-2 switching fabric [9]. The public peering at the IXP then becomes simpler due to the availability of physical infrastructure, with member ASes A and B (as shown in figure 2.2) initiating a BGP session to exchange packets through the IXP switch. On the other hand if E needs to send data to F, it requires the set up of BGP sessions between routers in the Internet cloud for it to be able to successfully transfer data to F. Figure 2.2 shows a scenario with the ASes peering at the

IXP switch. In this case, data sent between these ASes need not traverse the entire Internet and can be directly shared through the IXP. These peering links reduce transmission delays, use lesser international bandwidth and thus reduce overall costs of exchanging data for every IXP member AS.

The question arises as to when should an AS subscribe to an IXP? It is dependent on a variety of factors, primarily economic in nature. In the scenario shown in figure 2.2, if there is a significant volume of daily traffic between AS E and F, then it would probably be better off for F to peer at the IXP. Assuming both are stub ASes, the amount both would have to pay their respective transit providers would be far greater than the cost of setting up a peering link at the IXP. Data transfer costs, which in turn is dependent on traffic volumes are generally the determining factors behind AS peering at IXPs.

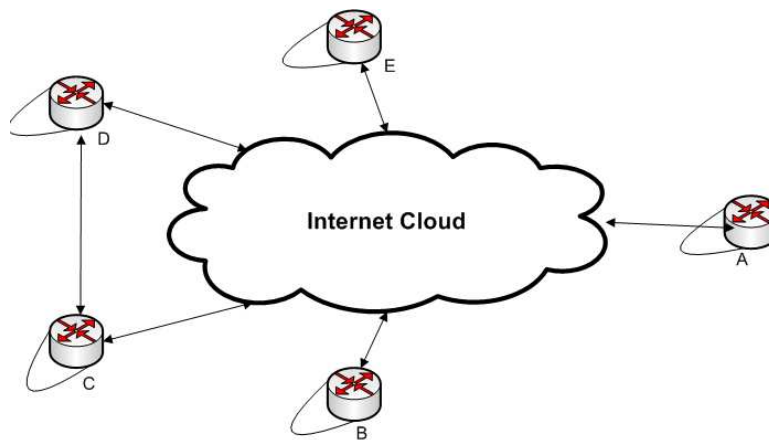


Figure 2.1 A set of ASes transmitting data to each other through the Internet. AS C and D share data through a direct peering link.

The advantages of peering at IXPs has led to a significant growth in the number of ASes peering at these switching points worldwide. As more and more ASes start peering

there is a greater percentage of data packets being routed in the Internet through these switches. In the following section we conduct some measurements and show that almost thirty percent of all routes in the Internet traverse an IXP. This leads to a greater number of peering links being formed at the IXPs thereby affecting the various characteristics of the Internet topology.

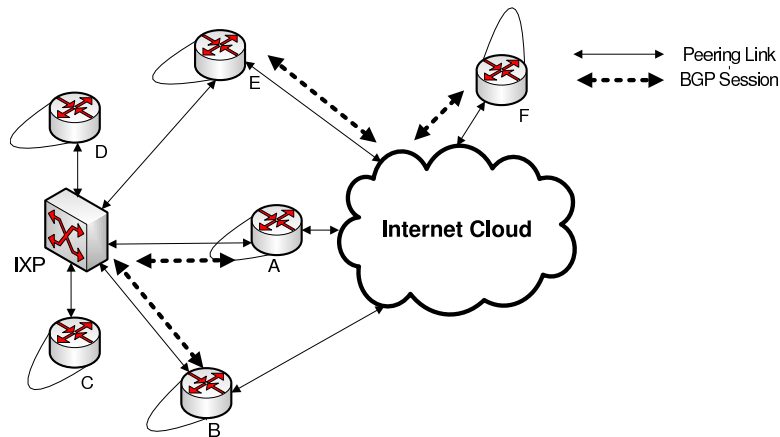


Figure 2.2 A set of ASes peering at an IXP. A and B set up a BGP session to exchange data while E and F use the Internet cloud to transmit data to each other. Any AS peering at the IXP may initiate BGP session with a peering AS.

2.2 IXP Growth

An increasing number of IXPs are being deployed across the world to enable more efficient traffic delivery over the Internet. This growth in the number of IXPs has been skewed with regard to the geographical location of these new IXPs being set up. There are numerically higher number of IXPs in Europe and North America than those in Asia or Africa for

example. However, there is no denying the fact that with an increasing number of IXPs coming up and with more ASes peering at these IXPs, the net Internet traffic going through these IXPs has increased over the years.

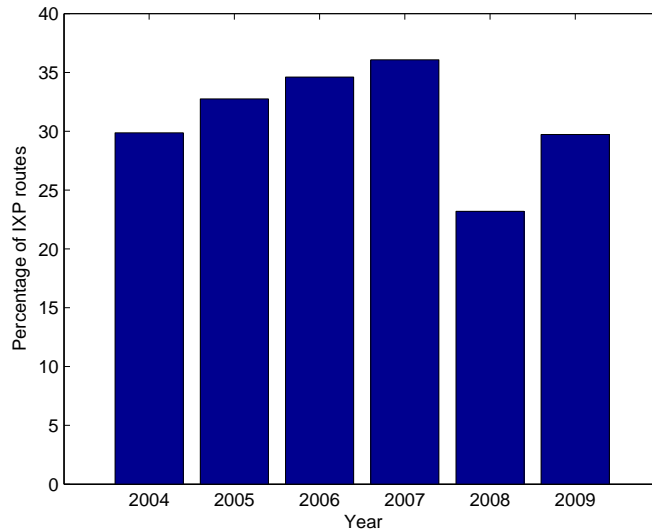


Figure 2.3 Percentage of IXP routes visible in one cycle of Skitter traceroute data every year for the month of September.

To study the impact of IXP routes we first need to quantify the percentage of routes going through any IXP in the Internet. To do this, we obtain one complete cycle of Skitter (now renamed Ark) traceroute data from the year 2004 to 2009 for the month of September. A complete cycle of data represents different skitter vantage points across the world sending out traceroute probes to the standard CAIDA destination list and records the paths taken. Based on the available list of IXP prefixes obtained from PCH and PeeringDB, we search for routes consisting of hops within these prefixes. An IXP route is thus defined as a route which contains atleast one hop through the network with a known IXP prefix. We count the

number of IXP routes obtained within one cycle and calculate its percentage based on the total number of routes obtained for the same cycle period. Figure 2.3 presents the percentage of IXP routes obtained every year and we observe that for most years we have at least 30 percent of all routes traversing an IXP. This means that almost one in every three routes goes through an IXP. The drop in percentage in 2008 and 2009 can be attributed to the fact that CAIDA's skitter architecture underwent a major change that year transferring to the Ark architecture. This resulted in a fewer traceroute probes being sent out and thus there were lesser routes recorded during this time. Table 2.1 presents the total number of routes observed along with the total number of IXP routes obtained. Oliveira et al. [10] point out that a high number of links and routes are not visible in the Skitter data due to its shrinking probing scope. The number of routes visible have decreased which has led to a decrease in the number of IXP routes too, but it still shows a significant percentage of routes being taken going through an IXP thereby underlying the importance of IXPs in the evolution of the Internet ecosystem.

2.3 Data sources and identifying IXP peering links from traceroutes

Internet topology evolution is typically studied by using various established datasets made available to the research community. BGP routing table dumps from the University of Oregon's RouteViews project [7] is the most extensively used resource. AS links appearing in the BGP tables represent existing links with a high probability of being alive and is

thus a more reliable source of information. However, if a link breaks or a node is down, the information takes some time to be updated through the network through BGP updates thereby leading to higher routing table convergence times. These updates have also been used as topology snapshots since they show a greater number of AS links over time [11].

Table 2.1 IXP growth obtained from searching known IXP prefixes from one cycle of Skitter data for the month of September

Year	IXP Routes found	Total routes visible	Percentage
2004	6963592	23312823	29.87
2005	6999045	21370051	32.75
2006	6387175	18455760	34.60
2007	5606309	15541716	36.07
2008	1629327	7020300	23.20
2009	1906532	7407891	25.73

Another widely available source of data is the data released by CAIDA under the Archipelago (Ark) infrastructure for research use [12]. From various vantage points across the Internet, ICMP probe packets are sent to a set of destination IP addresses using the traceroute tool. iPlanes [13] and Dimes [14] are other important and widely used sources of data publicly available for use in the study of Internet topology evolution.

There is a limited availability of data with respect to IXPs. PCH [1] maintains and makes available a set of BGP tables collected from a set of IXP routers worldwide while PeeringDB [15] is another project where IXP information is manually updated by individual providers. The recent IXP mapping effort by Augustin et al. [16] present IXP specific datasets including IXP IDs and network prefixes. Using a variety of tools developed, the authors come up with a list of IXP members and a set of peering links at these IXPs. They successfully discover and validate the existence of 44K IXP peering links which is roughly 75% more than reported in previous studies [6, 9]. This additional dataset of peering links at IXPs is used in this paper to create a more complete Internet topology graph.

IXP peering links have been mentioned as the hidden links which may be the key to solving ([6, 9]) the well known missing link problem in the study of Internet topology evolution. Table 4.1 presents a summary of the various data sets used and the nomenclature used throughout this paper.

Identifying IXPs in a traceroute has been described extensively in [6] and [17]. IXPs are assigned an IP address block and each AS peers at the IXP with a definite IP address for the interface within the given block. The list of IXP address blocks are available at PCH [1] and PeeringDB [15]. With the known list of IXP address prefixes we can search for every prefix from traceroute data and identify routes which include an IXP hop. As stated in [6] AS participants may then be identified by following the sequence of IP addresses before and after the known IXP address. By mapping the IP address of the participants to their AS

numbers we can obtain the participants at that particular IXP. We use these techniques to identify paths traversing an IXP in a later section.

2.4 Routing Performance

Our work is built on techniques and lessons from a variety of Internet measurement studies dealing with latency prediction, topology evolution and peering dynamics. In addition, statistical modeling methods such as generalized linear models help us identify and validate learning models from data generated by our measurement framework.

iPlane [13, 18] is the primary reference point in this work as we use publicly available datasets extensively to carry out our inter-AS latency estimates. The iPlane system continuously measures and maintains an annotated map of the Internet which is used to predict paths (both at the router level and the AS level) between arbitrary end nodes on the Internet. iPlane in itself is based on the broad vision of Clark et. al. [19] who discussed the need for an Internet wide knowledge plane which builds and maintains models of network functionality.

While latency measurements for paths utilizing an IXP have not been looked into in great detail, peering as a plausible source for the creation of quicker detour paths is mentioned by Zheng et. al. in [20]. The authors argue that routing policies impact RTTs directly for both intra- and inter-domain routing naturally giving rise to Triangle Inequality

Violations¹ (TIVs) in the Internet delay space. TIVs have been studied ([21]) and used extensively in overlay routing architectures [22] to implement efficient routing schemes for a variety of applications. For example, Ly et al. [23] use them to obtain latency reduction in popular online games.

2.5 Internet Topology

Internet topology evolution research is traditionally carried out with active measurements with [24] being one of the earliest works constructing topology snapshots from BGP routing tables and updates. This led to the general technique of constructing AS or router-level graphs of the Internet topology using both traceroute and BGP data [25, 26, 27] and analyzed these graphs based on various graph theoretical metrics. The focus has mostly been on designing measurements to maximize the number of links uncovered and solve the incompleteness problem [28, 9]. Researchers have all along concentrated on finding new links [29] and removing the expired links [10] formed due to the constantly changing Internet dynamics. Topology evolution needs to be studied in detail to help in the design and implementation of better topology generators and evolution models. These topology generators play a major role as newer and more efficient routing architectures can only be designed when effective topology maps can be created. Models proposed in [30, 31] aim to generate graphs which exhibit desired graph characteristics of the Internet.

¹Here the default latency between two arbitrary end-hosts is greater than the computed latency between them via another intermediate host

IXPs were recently identified as an integral component of the Internet architecture and were made a focal point of the study in [17] and [16]. He et al. [29, 6] carry out significant studies on uncovering IXP peering links and suggest that these locations hold the key to solving the *hidden links* problem in Internet topology research. By using the very comprehensive study carried out by Augustin et al. in [16] we aim to measure the impact these IXPs peering links are having on the evolving Internet topology today. Gregori et al. in [32] present an initial work discussing the impact IXP links are having on the AS-level Internet topology while we provide a more in-depth analysis and characterization of various graph based topology metrics in our work. Our aim is to interpret and analyze the effects these IXP peering links are having on the Internet topology.

2.6 Triangle Inequality Violations

Internet TIVs have been extensively studied in recent years with Zheng et al [20] reporting the correlation between inter-domain routing policies of ASes and the formation of TIVs. The authors surmise peering policies between ASes could lead to alternate shorter paths and hence more instances of a violation occurring. Savage et al. [33] first show the existence of **detour** routes, other paths through an intermediate host but to the same destination creating a TIV. The authors show that more than 30% of all default routes have a better detour path. The best detour paths more often have only one intermediate hop as shown in [22] which means the identification of TIVs do not require computing longer detours across

numerous other nodes. Lumezanu et al. in [34] analyze many new real world data sets of varying sizes and granularities to show that the TIVs are not just measurement artifacts and that their numbers could vary over time. Another section of inequality violation analysis has been in the performance analysis of network coordinate systems such as Vivaldi [35] and [21]. Due to the metric nature of these coordinate systems, TIVs cannot be replicated by the node embeddings. Wang et al. [21] identify these problems in the neighbor selection process and propose an alert mechanism eliminating the severe violations.

All the prior work analysing TIVs consider any Internet path across arbitrary end-hosts/ASes. In our work we carry out these study only for those paths traversing an exchange point. These exchange points paths have different characteristics because of the very nature of peering in the Internet, bypassing the transit providers and creating newer peering links between participating ASes. Analyzing these specific TIVs created due to exchange points have not been carried out in earlier work. IXPs in general and their effects on the topological evolution of the Internet have been a recent focus of the community. He et al. [6] suggest that the exchange points hold the key to solving the hidden links problem in Internet topology research, the primary goal of which is to uncover the maximum number of inter-AS links. Augustin et al in [16] carry out a comprehensive study in finding these IXP links and are successful in obtaining almost 18K previously unseen links. IXPs have been accepted to be an integral part of the Internet ecosystem and are playing a major part in the Internet interconnection dynamics today.

As identifying TIVs on a large scale is a computationally involved problem, we use parallel computing to carry out this process and harness the parallel processing capabilities readily available to most users. By breaking down the steps of the computation at hand we observe that the pattern matching and APSP graph algorithms may be implemented in parallel. This parallel implementation is carried out both on a multi-core CPU and on the GPGPU with NVIDIA's CUDA API. Huang et al. [36] propose a GPU based multiple pattern matching algorithm while the authors in [37] evaluate and implement a signature matching scheme on an Nvidia G80 GPU which outperforms a serial implementation on a Pentium4 by up to 9x. In our work we use the PFAC library which uses a variant of the well-known Aho-Corasick [38] algorithm. GPUs have also been used in solving various graph problems with Harish et al. [39] first using CUDA to compute the APSP on a graph. Katz et al in [40] improve this solution to give faster speedup results while Buluc et al in [41] implement a recursively partitioned APSP algorithm and obtain a very high degree of speedup. We use the technique proposed in [41] to implement our instance of APSP and explain the process in detail in the following sections.

CHAPTER 3: RELATED WORK

3.1 Routing Performance

Our work is built on techniques and lessons from a variety of Internet measurement studies dealing with latency prediction, topology evolution and peering dynamics. In addition, statistical modeling methods such as generalized linear models help us identify and validate learning models from data generated by our measurement framework.

iPlane [13, 18] is the primary reference point in this work as we use publicly available datasets extensively to carry out our inter-AS latency estimates. The iPlane system continuously measures and maintains an annotated map of the Internet which is used to predict paths (both at the router level and the AS level) between arbitrary end nodes on the Internet. iPlane in itself is based on the broad vision of Clark et. al. [19] who discussed the need for an Internet wide knowledge plane which builds and maintains models of network functionality.

While latency measurements for paths utilizing an IXP have not been looked into in great detail, peering as a plausible source for the creation of quicker detour paths is mentioned by Zheng et. al. in [20]. The authors argue that routing policies impact RTTs directly for both intra- and inter-domain routing naturally giving rise to Triangle Inequality

Violations¹ (TIVs) in the Internet delay space. TIVs have been studied ([21]) and used extensively in overlay routing architectures [22] to implement efficient routing schemes for a variety of applications. For example, Ly et al. [23] use them to obtain latency reduction in popular online games.

3.2 Internet Topology

Internet topology evolution research is traditionally carried out with active measurements with [24] being one of the earliest works constructing topology snapshots from BGP routing tables and updates. This led to the general technique of constructing AS or router-level graphs of the Internet topology using both traceroute and BGP data [25, 26, 27] and analyzed these graphs based on various graph theoretical metrics. The focus has mostly been on designing measurements to maximize the number of links uncovered and solve the incompleteness problem [28, 9]. Researchers have all along concentrated on finding new links [29] and removing the expired links [10] formed due to the constantly changing Internet dynamics. Topology evolution needs to be studied in detail to help in the design and implementation of better topology generators and evolution models. These topology generators play a major role as newer and more efficient routing architectures can only be designed when effective topology maps can be created. Models proposed in [30, 31] aim to generate graphs which exhibit desired graph characteristics of the Internet.

¹Here the default latency between two arbitrary end-hosts is greater than the computed latency between them via another intermediate host

IXPs were recently identified as an integral component of the Internet architecture and were made a focal point of the study in [17] and [16]. He et al. [29, 6] carry out significant studies on uncovering IXP peering links and suggest that these locations hold the key to solving the *hidden links* problem in Internet topology research. By using the very comprehensive study carried out by Augustin et al. in [16] we aim to measure the impact these IXPs peering links are having on the evolving Internet topology today. Gregori et al. in [32] present an initial work discussing the impact IXP links are having on the AS-level Internet topology while we provide a more in-depth analysis and characterization of various graph based topology metrics in our work. Our aim is to interpret and analyze the effects these IXP peering links are having on the Internet topology.

3.3 Triangle Inequality Violations

Internet TIVs have been extensively studied in recent years with Zheng et al [20] reporting the correlation between inter-domain routing policies of ASes and the formation of TIVs. The authors surmise peering policies between ASes could lead to alternate shorter paths and hence more instances of a violation occurring. Savage et al. [33] first show the existence of **detour** routes, other paths through an intermediate host but to the same destination creating a TIV. The authors show that more than 30% of all default routes have a better detour path. The best detour paths more often have only one intermediate hop as shown in [22] which means the identification of TIVs do not require computing longer detours across

numerous other nodes. Lumezanu et al. in [34] analyze many new real world data sets of varying sizes and granularities to show that the TIVs are not just measurement artifacts and that their numbers could vary over time. Another section of inequality violation analysis has been in the performance analysis of network coordinate systems such as Vivaldi [35] and [21]. Due to the metric nature of these coordinate systems, TIVs cannot be replicated by the node embeddings. Wang et al. [21] identify these problems in the neighbor selection process and propose an alert mechanism eliminating the severe violations.

All the prior work analysing TIVs consider any Internet path across arbitrary end-hosts/ASes. In our work we carry out these study only for those paths traversing an exchange point. These exchange points paths have different characteristics because of the very nature of peering in the Internet, bypassing the transit providers and creating newer peering links between participating ASes. Analyzing these specific TIVs created due to exchange points have not been carried out in earlier work. IXPs in general and their effects on the topological evolution of the Internet have been a recent focus of the community. He et al. [6] suggest that the exchange points hold the key to solving the hidden links problem in Internet topology research, the primary goal of which is to uncover the maximum number of inter-AS links. Augustin et al in [16] carry out a comprehensive study in finding these IXP links and are successful in obtaining almost 18K previously unseen links. IXPs have been accepted to be an integral part of the Internet ecosystem and are playing a major part in the Internet interconnection dynamics today.

As identifying TIVs on a large scale is a computationally involved problem, we use parallel computing to carry out this process and harness the parallel processing capabilities readily available to most users. By breaking down the steps of the computation at hand we observe that the pattern matching and APSP graph algorithms may be implemented in parallel. This parallel implementation is carried out both on a multi-core CPU and on the GPGPU with NVIDIA's CUDA API. Huang et al. [36] propose a GPU based multiple pattern matching algorithm while the authors in [37] evaluate and implement a signature matching scheme on an Nvidia G80 GPU which outperforms a serial implementation on a Pentium4 by up to 9x. In our work we use the PFAC library which uses a variant of the well-known Aho-Corasick [38] algorithm. GPUs have also been used in solving various graph problems with Harish et al. [39] first using CUDA to compute the APSP on a graph. Katz et al in [40] improve this solution to give faster speedup results while Buluc et al in [41] implement a recursively partitioned APSP algorithm and obtain a very high degree of speedup. We use the technique proposed in [41] to implement our instance of APSP and explain the process in detail in the following sections.

CHAPTER 4: INTERNET TOPOLOGY EVOLUTION

It has been suggested by the authors in [6] that the extra peering links at IXPs may hold the key to solving the *missing links* problem for the AS-level Internet and [16] shows that this hypothesis is probably true. However, the task ahead of us does not stop at uncovering these peering links. These additional links obtained need to be studied and analyzed in detail with respect to the existing Internet topology and their effects measured before a final conclusion can be arrived at. Any number of questions arise: Do the extra IXP links uncovered have a significant effect on the growing topology dynamics of the Internet? If the effects of these links are significant then how do we change our outlook in conducting topology research to accommodate these newer changes? Does solving the hidden links problem with these newer IXP links actually mean that we can accurately predict the growth of the Internet and verify previous evolution models as correct or not?

This chapter presents our study of AS visibility at IXPs with the primary aim of establishing the role of these IXPs in determining the evolving Internet topology. We try to find out if IXP data presents significant connectivity information not present in the more conventional data sources and detail our methodology of studying the effects of IXPs on the evolving Internet topology [42, 43]. We present our measurement details and datasets and follow it up with our graph-based analysis, results of which are presented here.

4.1 AS graph analysis

In this section, we present our methodology to obtain AS information from the different datasets we choose to consider. Our main aim is to identify the set of ASes visible, the number of AS links visible and other important network metrics representing important properties of the resultant graph. We look at topology metrics considered by Mahadevan et al. in [26] as they appear to fundamentally characterize Internet AS topologies and have been widely used.

Table 4.1 Datasets analysed and nomenclature

Dataset source	Name
RouteViews BGP [7]	<i>RVIEWS</i>
CAIDA (Ark/Skitter) [12]	<i>CAIDA</i>
Packet Clearing House [1]	<i>PCH</i>
DIMES [14]	<i>DIMES</i>
IXP Mapping [16]	<i>IXPMAP</i>
<i>RVIEWS + CAIDA + DIMES + IXPMAP</i>	<i>IXPALL</i>

As this study is primarily meant for comparison purposes, we decided to obtain a snapshot of Internet topology data from the data sources for a period of 31 days in October 2009. A month's worth of data provides a reasonable snapshot of the evolving Internet topology with enough time for different ASes and links to either show up or go down. We

obtain AS-level graphs from each data source as mentioned next and merge the 31 daily graphs into one graph per dataset.

4.1.1 Graph Construction

Table 4.2 Comparing the number of observed links in the *PCH*, *RVIEWS* and *IXPMAP* graphs

Dataset source Name	Links
<i>PCH</i> only (G_P)	370
<i>RVIEWS</i> only (G_B)	1408
<i>IXPMAP</i> only (G_M)	47507
<i>PCH</i> + <i>RVIEWS</i> ($G_P \cap G_B$)	71284
<i>PCH</i> + <i>IXPMAP</i> ($G_P \cap G_M$)	57
<i>RVIEWS</i> + <i>IXPMAP</i> ($G_B \cap G_M$)	159
<i>PCH</i> + <i>RVIEWS</i> + <i>IXPMAP</i> ($G_P \cap G_B \cap G_M$)	4250

RouteViews [7] collects and archives static snapshots of BGP routing tables from a set of monitors which can be accessed from the RouteViews data archives. Deriving the graphs from October 2009 we obtain a set of AS paths which we then convert to a set of AS links. The unique AS links obtained are set aside from which every individual AS visible is

then recorded. The final combined monthly graph we refer to as the *RVIEWS* graph in the rest of the paper.

CAIDA's IPv4 Routed /24 topology dataset [12] uses *team-probing* to distribute the work of probing the destinations among the available monitors using the *scamper* tool and forms a part of the Archipelago (Ark) topology infrastructure (which was formerly known as *Skitter*). Scamper probes are currently sent to a random destination prefix from a set of 7.4 million prefixes. As specified in [26] private ASes generate indirect links which we filter out during creation of the AS-level graphs and are then combined to form the final *CAIDA* graph.

PCH [1] releases the BGP routing tables at various IXP routers (currently 63) from various locations around the world. These routing table formats are the same as the RouteViews tables and hence are analyzed using a similar technique. We construct the *PCH* graph from these daily graphs.

The *DIMES* Internet mapping project is a distributed technique carrying out traceroute measurements from individual users located worldwide. Millions of traceroute/ping measurements are carried out by the low footprint DIMES agents installed on volunteer local hosts to present a detailed view of the Internet with a significant percentage of new links compared to those found in *RVIEWS* and *CAIDA*.

The IXP Mapping project [16] releases data specific to IXPs across the Internet with only peering links unearthed at these IXPs. We term this dataset *IXPMAP*. This is the

most comprehensive set of peering links present at IXPs currently available to the research community and we make it the primary source of study in this paper.

The peering links in *IXPMAP* are however not useful by themselves as they do not in any way give a complete picture of the Internet. As in other similar topology related studies, we combine these peering links with the other views of the Internet we obtain from the different datasets available to us. As stated earlier, we have the *CAIDA* traceroute based dataset (representing the data plane) and the *RVIEWS* BGP based dataset (representing the control plane). We compare the links obtained from the *PCH* data with the other BGP based dataset (*RVIEWS*) and present the result in table 4.2. It is observed from the table that *PCH* contains only 370 unique links in comparison to *RVIEWS* and the other IXP-specific dataset with a high number of links (almost 71k) being common among the BGP based datasets. The reasoning behind such similarity between these datasets is the fact that both are derived from BGP tables at a set of routers some of which are actually common to both sources. Due to such a characteristic of the *PCH* data we simply combine the unique links obtained from this dataset to the *RVIEWS* graph to simplify our analysis and reduce the number of graphs generated to three.

We complete the entire picture of the Internet by combining *CAIDA*, *RVIEWS*, *DIMES* and *IXPMAP* to one entire *IXPALL* graph. This graph is characterized by the data plane (*CAIDA*), the control plane (*RVIEWS*), extensive peer to peer links (*DIMES*) and the peering links (*IXPMAP*) and built over a one month period, is relatively representative of the Internet during that period of time.

4.1.2 Validity of chosen datasets

As detailed in the subsection above, we carry out a careful consideration of each of the available datasets before combining them to create the final combined graph of the Internet. While each of the links made available are validated by the sources before release, it can be considered that over time some of the links may simply expire and new ones created. This is specially true for the *IXPMAP* dataset which is not maintained by the original developers any more. However we do not consider the dataset to have become corrupt and rendered useless. By using historical data (from *CAIDA* and *RVIEWS*) for that particular month we obtain a relatively clear and correct snapshot of the Internet for that particular period and study the graphs. The question of the current validity of the peering links could be raised when the topology evolution is being studied over an extended period of time, something which is not the goal in this work. The IXP peering links would have a high probability of remaining valid for the period considered and thus enable an accurate study of their effects on the AS-level topology of the Internet.

We carry out graph based comparison studies in the next section between *CAIDA*, *RVIEWS* and the *IXPALL* datasets and do not report the results of the *DIMES* dataset individually. This is because both *CAIDA* and *RVIEWS* present distinctly different views of the Internet as mentioned earlier (the data and control planes respectively) while *DIMES* presents an overall view based on the locations of the user agents. However, the unique links from *DIMES* are used in creating our view of the complete Internet in *IXPALL*.

There are two primary reasons we combine the IXP links with (G_P) and not the skitter or routeviews derived graphs:

- Not present in BGP tables: As pointed out in [44] most IXP links are not visible in BGP routing tables. By combining these extra peering links with BGP links we would get a more complete graph with little overlap between the two merged graphs. Table 4.2 presents a comparison of the number of unique links visible at the *PCH*, *BGP* and the *IXPMAP* graphs. We observe that a high number of links (47k) are visible only in G_M while those in both G_P and G_M are less than G_B and G_M . To reduce overlap as much as possible and for better comparisons, merging G_P with G_M seems to be a better option.
- Routers at IXPs: G_P is constructed from the BGP tables at routers situated at specific IXP locations worldwide. However, when there is no significant difference in the number of links visible at these routers than the *BGP* dataset, we believe adding the links to the graph derived from these locations would be a more suitable recreation of the existing topology.

4.2 Topology characteristics

4.2.1 Degree distribution

The node degree distribution is the probability distribution of the node degrees in a graph. In other words, it is the probability that a node selected randomly is of k -degree and this probability is calculated by: $P(k) = \frac{n(k)}{n}$, where $n(k)$ is the number of k -degree nodes in a graph with total number of nodes n . Scale-free networks such as the Internet have been shown to exhibit power law degree distributions [45] and hence the power law exponent is computed for this metric. This power law model has had a significant effect on Internet topology research and topology generators ([31],[46]) are designed primarily adhering to this characteristic.

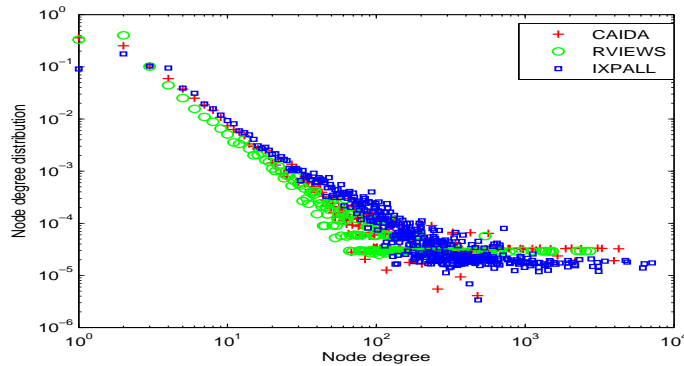


Figure 4.1 Node degree distribution. Power law behavior remain evident for all three datasets.

From figures 6.7(a) and 6.7(b) we observe distinct power law characteristics being followed by all three topology datasets for a wide range of node degrees. The average node degrees (listed in table 4.3) are in \bar{k} -order with $RVIIEWS \leq CAIDA \leq IXPALL$ and

the average node degree in *IXPALL* exhibiting a significantly higher value than the others. This is largely due to popular IXP nodes exhibiting high degrees due to multiple peering ASes at one location. The power law exponents computed are not affected significantly by these additional high degree nodes with the γ value for the combined *IXPALL* graph being slightly higher than the others (refer to 4.3 for complete details). The authors in [26] point out that a natural cut-off at power-law maximum degree is obtained at: $k_{max}^{PL} = n^{\frac{1}{(\gamma-1)}}$. From table 4.3 we observe that the maximum node degree k_{max} for the *IXPALL* is closest to the power law thereby meaning that the power law approximation for this set is relatively accurate.

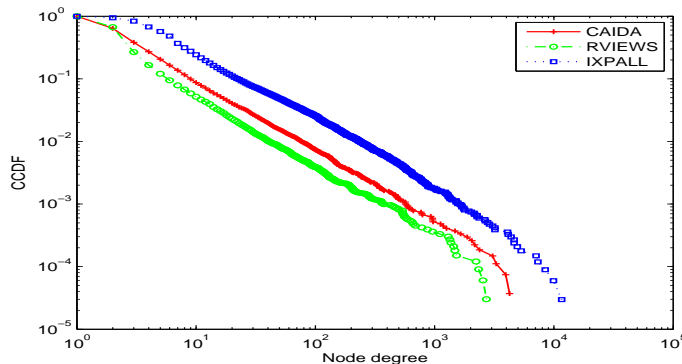


Figure 4.2 CCDF of node degree distribution for the three datasets. Power law behavior remains consistent across datasets.

This result shows that the degree distribution of the *IXPALL* graph still does follow a power law but with different parameters. By uncovering of these new peering links at IXPs the basic topology evolution characteristic of the Internet does not deviate from the existing power law characteristic and its behavior remains the same. The CCDFs of these graphs

also reiterate this conclusion. The addition of an extremely high number of unique peering links does not break the power law characteristics of the graph. Figure 6.7(b) shows that the *IXPALL* graph has a greater number of nodes for corresponding node degrees in comparison with the *CAIDA* and *RVIEWS* graphs. This is simply due to the fact that a high number of low to medium degree ASes (degrees of 10 to 1000) peer at the IXP switches with each other. The newer links uncovered are between these peering ASes increasing the total number of ASes with these degree characteristics. However it is evident from the figure that the net characteristic of the Internet's degree distribution still remains the same even with the addition of the IXP peering links.

4.2.2 Power law degree distributions

The now famous paper by Faloutsos et al. [45] exhibiting a power-law degree distribution of the Internet graph at the router level led to a plethora of research in this evolution characteristic of the Internet. Suggested scale-free network models based on preferential attachment [47] describe the power law degree distributions with an exponent α between 2 and 3. However there has been a large amount of follow up work where the degree distribution characteristic has been shown to be a result of an inherent bias of traceroute based measurement mechanisms. Lakhina et al. in [48] show that traceroutes from a small set of sources to a larger set of destinations measure edges in a highly biased manner with the degree distribution results differing sharply from that of the actual underlying graph. Achlioptas et

al. in [49] provide a mathematical proof of the results obtained in [48] while a recent work by Willinger et al. [50] discuss the origin and reasons behind the *scale-free Internet myth*.

We discuss this particular issue in this paper as in our first result we do show that the combined Internet graph exhibits the power-law distribution with an exponent of 2.18 (as listed in table 4.3). It has to be noted however that the basis for not supporting this power law characteristic is for traceroute based studies from a very small set of source monitors to thousands of destination IP addresses across the globe. The authors of [16] carefully select a large number of traceroute enabled looking glass (LG) servers (about 2300) from which they send out targeted traceroute probes to responding target hosts within (or a neighbor of) an AS peering at a known IXP prefix. We believe this technique will not be subject to the traceroute sampling biases as discussed in [48, 49] and the IXP peering links obtained also do not show such a property when analyzed in isolation. When combined with the other datasets to represent the entire Internet, these links end up affecting the graph properties but nearly not enough when node degree distributions are studied. The *IXP MAP* dataset is inherently free of the traceroute bias in our opinion thus making it beneficial for us to study its effects on the Internet topology. Moreover, the objective of this work is a complete understanding of the IXP link effects (and not only degree distributions), which we carry out for other important topology metrics.

4.2.3 Joint degree distribution

The joint degree distribution gives us an idea of the general *neighborhood* of a randomly chosen node with an average degree. The immediate one hop neighborhood of the node gives significant information not only about the interconnections between nodes but also the structure of the area around the node.

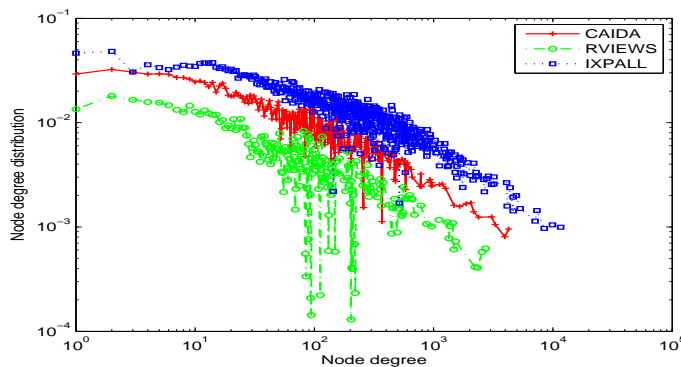


Figure 4.3 Normalized average neighbor connections. *IXPALL* comprises of excess tangential links connecting high degree nodes.

Mahadevan et al. in [26] define the joint degree distribution (JDD) as the probability that a randomly selected edge connects k_1 and k_2 -degree nodes: $P(k_1, k_2) = \frac{m(k_1, k_2)}{m}$, where $m(k_1, k_2)$ is the total number of edges connecting nodes of degree k_1 and k_2 . Figure 4.3 shows the JDD for the different graphs. Since *CAIDA* has the highest number of radial links connecting low-degree customer AS nodes to high-degree provider AS nodes, it is at the top for lower node degrees. Since *IXPALL* contains all these nodes and links from *CAIDA* its behavior is very similar initially. However the effect of IXP peering is evident for medium to high degree nodes (10 to 1000). Numerous peerings between ASes at different

locations worldwide result in tangential links between ASes of similar higher degrees resulting in the *IXPALL* graph showing consistently high values throughout the middle and latter sections of the graph. Figure 4.4 presents the ccdf of the average neighbor connections against average node degrees. A higher percentage of *CAIDA* nodes have an average neighbor degree greater than *RVIEWS* but the effect of the extra peering links added in *IXPALL* is not extensive when combined with the graphs. This is because only a small number of extra nodes with higher number of links are included, thereby not affecting the actual number of nodes. Thus we can accurately conclude that the peering links at IXPs again significantly affect the JDD of the Internet topology graphs obtained from the traditional sources.

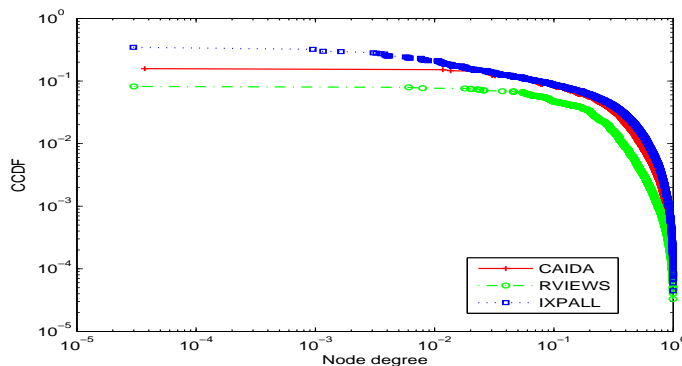


Figure 4.4 CCDF of average neighbor connections. *IXPALL* is not significantly different due to only a few number of high degree nodes being added to the *CAIDA* and *RVIEWS* datasets.

A summary statistic of the JDD is the average neighbor connectivity, the average neighbor degree of the average k -degree node. The average neighbor degree for the different graphs is listed in table 4.3. As seen in the degree distribution plots, *CAIDA* exhibits values

greater than the BGP based graphs but the IXP peering nodes have high average neighbor degrees, which has an overall effect in increasing the average degree of the neighbor nodes in *IXPALL*.

Another scalar value summarizing the JDD is the assortative coefficient [51] which measures mixing patterns between nodes. The coefficient r , which lies between -1 and 1 denotes the correlation between a pair of nodes, with negative values of r indicating relationships between nodes of different degrees and positive values of r showing that nodes have correlations between nodes of the same degree. With the scale free nature of Internet, it is not surprising to see all our graphs being disassortative in nature with a high number of radial links connecting nodes of different degrees [26]. Since the traceroute based studies are unable to find a high number of tangential links, all the graphs show higher disassortative trends. However the peering links in *IXPMAP* are the source of the tangential links between high degree nodes thereby resulting in a relatively higher assortative coefficient value.

4.2.4 Clustering coefficient

The value for the local clustering coefficient of a node denotes how close its neighbors are to forming a clique. This metric serves as a supplement to the JDD by providing more information about how the neighbors interconnect. If the average number of links between k -degree nodes is $\bar{m}_{nn}(k)$, then the local clustering coefficient $C(k)$ is (from [26]) : $C(k) = \frac{2\bar{m}_{nn}(k)}{k(k-1)}$. If two neighbors of a node are also connected, then it forms one triangle while a

triplet of nodes is formed when out of three nodes either two or three nodes are connected to each other. An open triplet is formed with two connections while a closed triplet is created when all the nodes are connected to each other. The global clustering coefficient is a percentage of the number of closed triangles (made up of three closed triplets) in the entire graph over the total number of triplets in the graph.

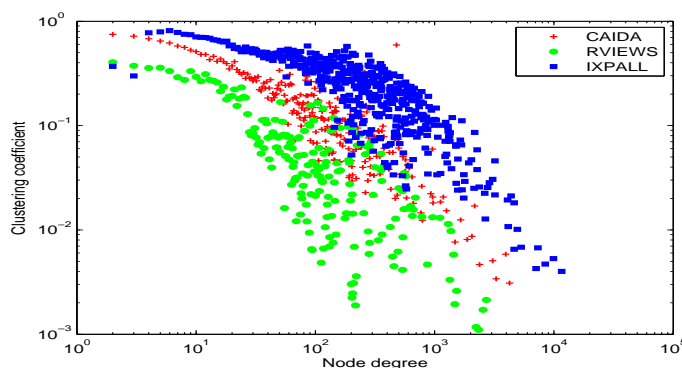


Figure 4.5 Local clustering with increasing node degrees. *IXPALL* exhibits constant high clustering values due to a high number of links being clustered at the IXP nodes.

From a high local clustering value of a node it can be inferred that its neighbors have greater interconnections which in turn leads to greater path variance. Such a characteristic would provide interesting ramifications for ASes peering at individual IXP locations. A pair of ASes would be more eager to peer if there is a potential to peer with other ASes already present at that location. With a high local clustering value, all ASes at the IXP would be able to transmit traffic to each other more efficiently through a subset of peering ASes. These highly clustered networks would also help in the routing performance under different conditions. From table 4.3 we observe *CAIDA* to have a higher mean clustering value but

IXPALL exhibits a clustering coefficient double that of the former. As mentioned in [26], this is due to greater differences in disassortativity and JDD values. In figure 4.5 we observe *IXPALL* exhibits high clustering values for lower degree nodes.

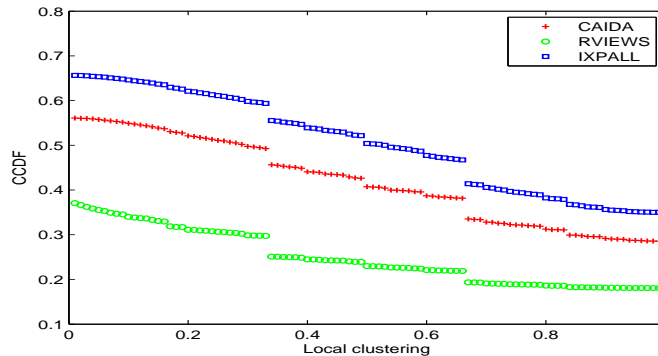


Figure 4.6 CCDF of local clustering values. *IXPALL* shows a consistently high probability for all clustering values considered.

These are due to the *CAIDA* nodes which are highly disassortative, meaning that lower degree nodes have a higher probability of being connected to high degree nodes. For higher degree nodes, the local clustering values are significantly higher. This is because the average node degree (\bar{k}) for *IXPALL* nodes is much greater in comparison. The ccdf of local clustering values (figure 4.6) obtained reinforce the above conclusions whereby there is always a higher probability of nodes exhibiting a particular local clustering value.

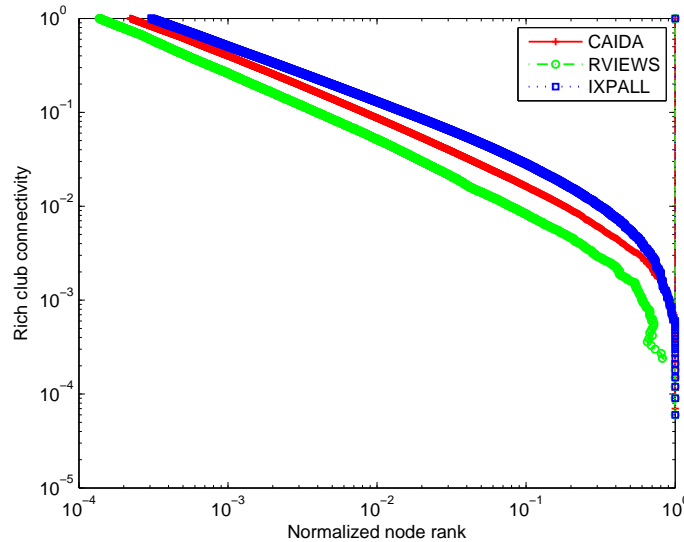


Figure 4.7 CCDF of the Rich club connectivity (RCC) for the three graphs. The highest connectivity among high degree nodes is in *CAIDA* while *IXPALL* high degree nodes are not connected between themselves.

4.2.5 Rich club connectivity

The Rich club connectivity (RCC) metric, introduced by Zhou and Mondragon in [31, 52] provides an insight into the properties of power law networks. *Rich nodes* are a small number of nodes with large numbers of links forming a core club of nodes which are very well connected to each other. As defined in [26], if $\rho = 1 \dots n$ is the first ρ nodes ranked in decreasing order of node degrees, then the RCC $\phi(\rho/n)$ is the ratio of the number of links in the subgraph induced by these ρ nodes to the maximum possible links $\rho(\rho - 1)/2$. It is pointed out in [31] that the RCC is a key component in characterising Internet AS-level topologies.

Figure 4.7 presents the RCC for the various graphs and it can be seen that *CAIDA* exhibits the highest RCC values. Even though *IXPALL* has a greater number of links its lower RCC means that the higher degree nodes are not connected extensively with each other. The subgraphs induced from these high degree nodes do not come close to forming cliques which can be explained from the location based nature of IXPs. IXPs in general are not connected to each other and the peering links created at these locations remain localized. These peering links denote a cooperation only between a pair of nodes which are independent of other peering links. The *IXPALL* graph would exhibit higher RCC values if more ASes at the IXP peer with a greater number of ASes already peering there. The potential for a greater IXP utilization is evident from this result as there is an opportunity for more ASes to come up with peering agreements and ensure even better connectivity.

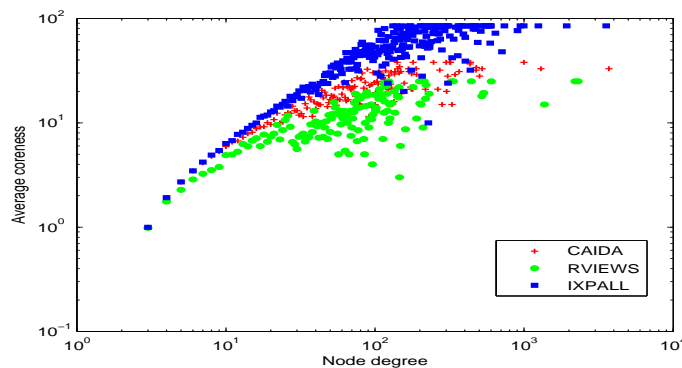


Figure 4.8 Average node coreness with increasing node degrees. The increase in coreness roughly follows a power law for all graphs for low and medium and degree nodes before becoming stable.

4.2.6 Node coreness

The authors in [26, 53] define the k -core of a graph as the subgraph obtained from the original graph by the iterative removal of all nodes of degree less than or equal to k . The node coreness (κ) can be defined as the highest k for which the node is present in the k -core but removed in the $(k + 1)$ -core. Thus all one degree nodes have coreness equal to 0 while the maximum node coreness κ_{max} is termed the *graph coreness*. In this case the κ_{max} -core of the graph is not empty but the $(\kappa_{max} + 1)$ -core is. The graph *fringe* is defined as the set of nodes in the graph displaying minimum coreness κ_{min} .

The node coreness is a more advanced version of node connectivity than the node degree as it tells us how well the node is connected to the entire graph. A node may have a high degree but its connectivity to other parts of the graph is dependent largely on its neighbors. The best example to describe this is a high degree hub of a star which has a coreness of 0 with its neighbors only having a very low degree (one), which when removed leaves the hub disconnected.

From table 4.3 we observe *IXPALL* exhibits significantly higher average node coreness ($\bar{\kappa}$) and maximum coreness (κ_{max}) values. The core size ratio is also higher indicating the general higher general connectivity due to IXP links induced in the graph. Figure 4.8 displays this result showing the effect of the IXP peering links increasing the overall coreness for nodes with all low, medium and high degrees. It is also evident from the figure that the increase in node coreness follows a power law increase for nodes upto degrees of 100

before remaining stable for higher degree nodes. Likewise the fringe size ratio is also the lowest in *IXPALL* which means fewer nodes with minimum coreness thereby leading to a better connected graph than the two others. The coreness result presents an important characteristic: the fact that the greater number of links also leads to better connectivity. These new links are not all only tangential links between low degree nodes but contain a generous amount of radial links leading to better node connectivity.

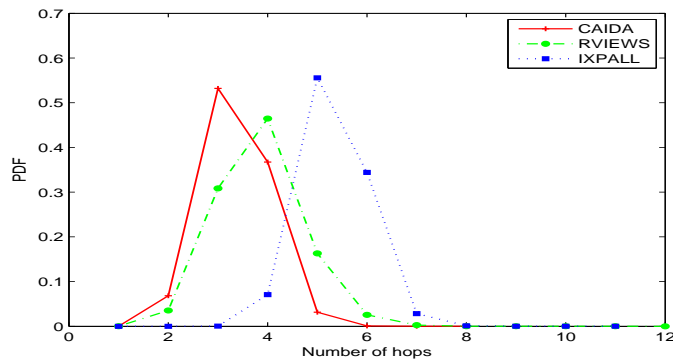


Figure 4.9 Distance distribution showing that *IXPALL* links increase the number of hops between two arbitrary hops over the Internet.

4.2.7 Distance and eccentricity

The **distance distribution** $d(x)$ is the probability for a pair of random nodes to be at a distance of x hops within each other whereas **eccentricity** is the maximum distance between the pair of nodes. Thus the maximum eccentricity in a graph is also the maximum distance and is termed the *graph diameter*. This metric is important while designing efficient routing

policies to enable paths with lesser hops to be chosen. The authors in [26] also point out that the distance distribution plays a major role in helping the network recover from virus attacks. Figure 4.9 presents the distance distribution values of the three graphs studied. We observe that about 55 percent of nodes in *IXPALL* are separated by a distance of 5 hops while it is lower for the other graphs.

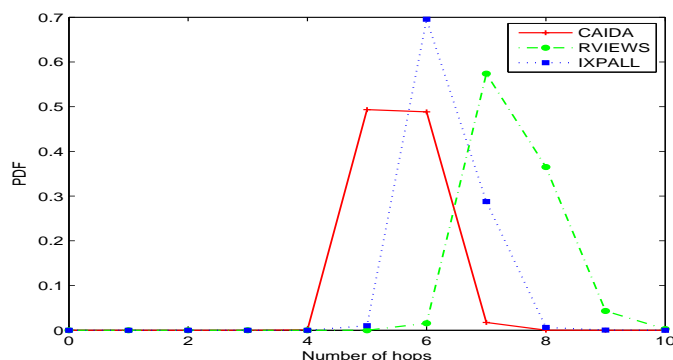


Figure 4.10 Eccentricity distribution of three graphs. Similar values are observed with *IXPALL* having the highest percentage of nodes separated by 6 hops between them.

Even though *IXPALL* has a greater number of links (which means that average distances should decrease), the average distance value is greater suggesting that deployment of IXPs do not decrease the path lengths between end-hosts on the Internet. There could be routing performance efficiencies through IXP deployment but the number of hops traversed largely remain the same. Figure 4.10 shows that maximum distances for a majority of the nodes are similar across all graphs with almost 70 percent of *IXPALL* nodes separated by a maximum of 6 hops from each other.

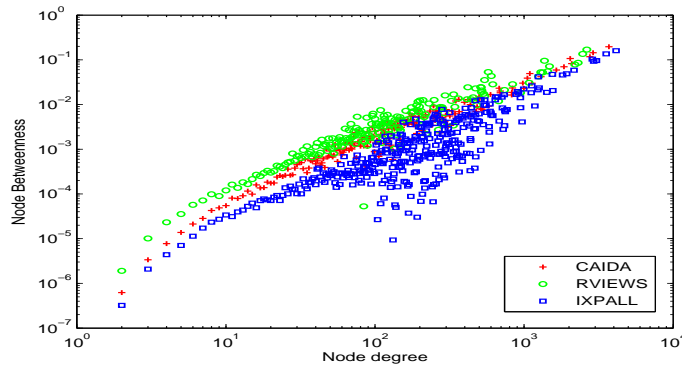


Figure 4.11 Normalized node betweenness with $n(n - 1)$ being the normalization factor.

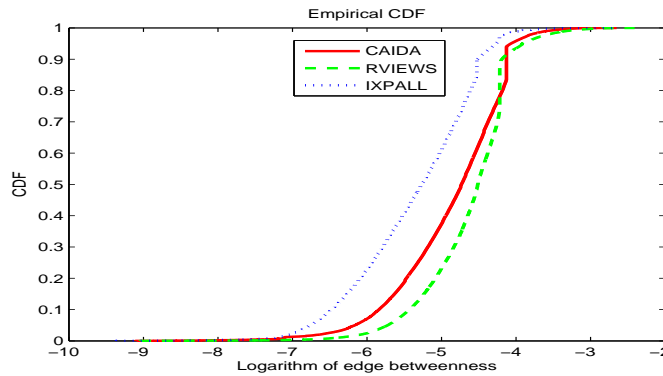


Figure 4.12 CDF of log of edge betweenness for the three graphs. *IXPALL* has the highest percentage of edges with the lowest edge betweenness values.

4.2.8 Betweenness

The most common and effective means of measuring node centrality is **betweenness**. Nodes which appear on a greater number of shortest paths between any pair of nodes in the graph exhibit a higher betweenness value. Such nodes are considered to be more *central* than others since it is assumed that majority of the traffic on a network is sent along the shortest path from source to destination. Potential traffic load on nodes/links may be estimated

from betweenness values of certain critical nodes which would also point to locations for potential congestion. Using a relatively quick algorithm [54] to calculate the betweenness centrality of the nodes, we obtain the normalized betweenness distribution with increasing node degrees. Since the maximum number of paths possible in a graph is $n(n - 1)$, all the graphs are normalized by this value and the results shown in figure 4.11. It can be observed that all three graphs exhibit a power-law function of node betweenness with increasing node degrees with *IXPALL* exhibiting lower values overall. Higher numbers of nodes of all degrees (mainly medium degrees) in *IXPALL* leads to greater path diversity. This means there is a presence of a greater number of nodes for paths of equal distance between all pairs of nodes leading to the lower betweenness values observed.

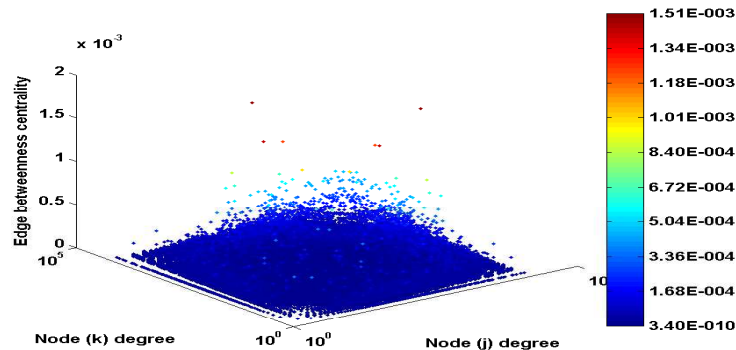


Figure 4.13 Scatter plot showing edge betweenness centrality for *IXPALL* for edges with different node degrees. Centrality values overall remain quite low.

Continuing from the node betweenness values exhibited in figure 4.11 we compute the normalized edge betweenness for the graphs and present the results in figures 4.12. The figure shows the CDF of the log of betweenness values for all edges in the three graphs.

It can be seen that *IXPALL* has the highest percentage of edges with the lowest edge betweenness values (as is evident in the scatter plot in figure 4.13 of edge centrality, with a high concentration of points with very low betweenness values). This means that a high percentage of IXP peering edges (along with nodes) do not fall on the available shortest paths between nodes in the entire graph. It has to be noted here that inter-domain routing in the Internet does not follow conventional shortest path approaches and is actually determined by inter-ISP routing policies and hot-potato routing in BGP. Betweenness can thus not be considered as an entirely accurate indicator of Internet path performance except to give an idea of the relative importance of the nodes/edges along a shortest path. We may conclude from this result that IXPs do not necessarily decrease the hop count of paths between ASes peering at those locations as path lengths essentially remain similar to other established paths from source to destination AS.

4.3 Analysis and discussions

Combining the extra peering links visible at the IXPs with the general structure of the Internet has given us a varied set of characteristics of the completed picture of the Internet (the data plane combined with the control plane and the peering links). Comparing the derived topologies based on the available graph metrics gives us an insight into the effects the peering links uncovered at the IXPs are having on the topology evolution of the Internet.

The most widely studied node degree distribution behavior of the Internet remains essentially unchanged even after the addition of all the peering links. The scale-free nature of the Internet graph, based on the different *views* considered, does remain the same. Numerous instances of related work have noted that the IXP peering links hold the key to solving the missing links problem and our findings suggest peering links provide a part of the solution to the problem. However it has to be mentioned that there has been work following the famous paper by Faloutsos et al. [45] which have discounted the scale-free nature of the Internet [50, 48, 49] due to inherent biases in the traceroute mechanisms.

Observing the effects of IXP peering links on other important metrics leads to some interesting insights. Higher JDD values for medium to high degree nodes means that well connected ASes (and providers, preferably the higher tier ISPs) are setting up peering relationships at exchange points. Such peering links lead to higher average neighbor degrees. A generous mix of both tangential and radial links are evident in the *IXPALL* graph unlike in *CAIDA* where there is a high number of radial links connecting nodes of vastly different degrees. The high JDD also comes with high levels of local clustering due to IXP peering links. This characteristic should and does directly serve to provide an incentive to ASes to peer at an IXP. A high number of links inevitably leads to greater local clustering but the RCC on the other hand displays the fact that there is little interconnection between the IXPs between themselves. Such connections between IXPs are however not needed since they are constructed to provide a platform for local interconnectivity amongst coordinating ASes.

The node coreness metric which points out how "deep in the core" the node is situated [26], shows that the nodes in the *IXPALL* graph are mostly well connected with well connected neighbors. The IXP substrate has thus become an important component of the Internet's infrastructure leading to a 'flatter' Internet from a hierarchical one. Gill et al. in [2] reported the changing characteristic of the Internet to a more "flat" architecture which can be inferred by the results obtained by us with the coreness metric. The greater number of peering links between ASes at IXPs lead to those ASes getting deeper into the core of the Internet with decreasing emphasis on connections with upper tier ASes. The authors in [55] and [56] have all pointed towards this evolution characteristic of the Internet and our coreness metric based result presents a theoretical confirmation of these observations.

Node and edge betweenness are two measures of centrality from which further inferences can be made about the effects of IXPs peering links. Both these metrics point towards lower values for *IXPALL* which means not many AS-AS peering links are a part of the shortest paths between ASes. Zheng et al. [20] show that routing policies and the layer 2 technology used on peering links may lead to cases of Triangle Inequality Violations (TIVs) [21, 34] in the Internet and not necessarily provide significant savings on RTT measurements between ASes. With most detour paths [57] forming TIVs, peering links do not necessarily lead to shorter paths along the Internet. The results we obtain again largely confirms this Internet path characteristic from a theoretical perspective.

Coming back to the questions we posed at the beginning of the paper we observe that IXP links indeed play a major role on topology characteristics of the Internet. Their

effects on various important topology metrics should make the Internet topology research community stand up and take notice of this integral component and give due attention to uncovering more peering links at IXP locations worldwide. While Augustin et al. in [16] present a first step in carrying out a comprehensive study to uncover peering links, there is no sustained effort in the community to continue such studies at the moment. On the other hand, the flattening of the Internet topology structure [2] shows the growing trend among the ASes to move away from higher tier transit ISPs towards creating inter-AS peering links. These characteristics and the incredibly high number of IXP peering links point towards the fact that IXPs are indeed the key towards solving the missing links problem and with their addition to the visible Internet topology we will go a long way to verifying the validity of topology generators and evolution models.

Table 4.3 Table detailing graph summary statistics

Metric	Property description	CAIDA	RVIEWS	IXPALL
Average Degree	Number of nodes (n)	26957	33199	33606
	Number of edges (m)	94161	77101	320728
	Average node degree (\bar{k})	6.98	4.64	19.08
Degree Distr	Max node degree (k_{max})	4249	2717	11623
	Power law max degree (k_{max}^{PL})	7301	9690	7809
	Exponent of $P(k)(-\gamma)$	2.14	2.13	2.16
	Maximum degree ratio	0.16	0.08	0.35

Continued on next page

Metric	Property description	CAIDA	RVIEWS	IXPALL
Joint degree distr	Avg neighbor degree ($\bar{k}_{nn}/(n-1)$)	0.028	0.015	0.019
	Assortative coefficient (r)	-0.16	-0.20	-0.07
Clustering	Mean clustering (\bar{C})	0.39	0.25	0.29
	Clustering coefficient (C)	0.02	0.01	0.04
Coreness	Average node coreness ($\bar{\kappa}$)	2.05	1.33	4.34
	Max node coreness (κ_{max})	38	25	87
	Core size ratio (n_{core}/n)	$3 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
	Minimum deg in core (k_{core}^{min})	75	38	119
	Fringe size ratio (n_{fringe}/n)	0.37	0.33	0.25
	Max degree in fringe	3	8	6
Distance	Average distance (\bar{d})	3.364	3.844	3.333
	Std deviation of distance (σ)	0.661	0.848	0.655
Eccentricity	Graph radius	4	6	5
	Average eccentricity (\bar{e})	5.522	7.443	6.291
	Graph diameter	8	11	9
Betweenness	Avg node betweenness	$8.78 \cdot 10^{-5}$	$8.57 \cdot 10^{-5}$	$6.96 \cdot 10^{-5}$
	Avg edge betweenness	$3.57 \cdot 10^{-5}$	$4.99 \cdot 10^{-5}$	$1.45 \cdot 10^{-5}$

CHAPTER 5: BANDWIDTH MEASUREMENTS

5.1 Bandwidth studies of popular web destinations

With IXPs being such an important component of the Internet infrastructure, their role in accessing popular websites and Internet services from end hosts has become quite significant. With a large percentage of Internet traffic now being accessed from wireless and mobile devices, there is a shift towards providing these services in an efficient and cost-effective manner. We continue our study of IXP effects by focusing on the paths' available bandwidth [58, 59]. While we earlier measured and analyzed general paths from various PlanetLab vantage points, in the second part of the study we concentrate only on the most popular websites, content distribution networks (CDNs) and cloud computing services. These popular destinations are accessed by millions of users worldwide on a daily basis or provide critical services on which these users depend on. A large percentage of Internet traffic is due to these popular web destinations for which we study IXP effects on these Internet routes. We now discuss the methodology and intuition behind the measurements we carry out along with a brief description of the various tools used in our studies.

5.1.1 Source and destinations

We carry out our experiments from a host at University of Central Florida (UCF) to the servers serving the top thousand websites across the world according to Alexa's ranking list as on March 1st, 2010. These popular websites were chosen for this initial study because they represent a large section of the Internet traffic and would largely be available with low down-time. Since our main intention is to study routes from the source to these destinations traversing an IXP, we conduct traces to these thousand websites and select the set of destinations whose path encounters at-least one IXP hop. Most of these popular websites are complemented by an array of CDNs at important geographic locations to reduce delivery latencies and increase reliability by providing a safeguard against delivery failures. Some of these popular websites are also hosted in the cloud and served by the popular cloud computing services. Video and music streaming applications (such as Internet radio sites Pandora, Spotify and so on) fall in the latter category of being served either by CDNs or hosted in cloud based systems. We group the diverse range of destinations monitored into two major groups:

1. World Wide Web (WWW): These comprise of the top 1000 websites according to Alexa's ranking list as mentioned. Regular websites, music, video providers and other streaming applications are all classified here.
2. Services: These comprise of the popular CDNs and cloud services which not only generate content but also provide services to many of the top websites selected.

After filtering out the set of destinations we conduct specific measurement experiments from the same source host at UCF as previously mentioned. The main reasoning behind using only one source is because studying the characteristics of IXP routes is a primary goal of our work and with peering at IXPs dependent on AS agreements, an IXP route is almost always unique to a given *(source AS, destination AS)* pair. Only sources within the same AS would exhibit similar paths whereas geographically co-located hosts have a high probability of not using a comparable path through the same IXP to the given destination. Thus the following measurements are performed from one source host at UCF unless otherwise mentioned, where a particular set of measurements are carried out from a number of geographically diverse Planetlab [60] vantage points.

5.1.2 Route characteristics studied

Our primary goal in this section is to gain an understanding of the existence of bottleneck locations on IXP routes and how the existence of these bottlenecks are affected by the IXP. We use the popular Pathneck active probing tool [61] to accurately locate the bottleneck hop/link and obtain available bandwidth information. Hu et al. in [62] term these bottleneck hops as *choke points* and the downstream link to the choke point as a *choke link*. We measure the *persistence* before identifying the bottleneck hops since this bottleneck location could change with changes in the underlying IXP path. Frequent link flapping is an undesirable characteristic of Internet routes which occur mainly due to the propagation of unstable BGP

routing information. Path stability is thus very important to determine bandwidth choke points and *persistence* is a popular metric aimed to quantify the stability of an Internet route. We follow it up with a bottleneck persistence study to identify if these hops are loaded only temporarily (which may be due to an external event) or not and the relation between bottleneck points with both packet loss and queuing delays occurring along the IXP route. We also measure the standard delay, loss and jitter metrics for the identified IXP routes.

5.1.3 Tools used

As mentioned previously, we have used Pathneck [61] to identify both route and bottleneck persistence in our measurements. Pathneck uses a recursive packet train (RPT) probing approach combining load and measurement packets in its probing mechanism to efficiently identify the hop limiting the available bandwidth on a path. We also use the Tulip probing tool [63] to detect the exact position of packet loss along the path and estimate queuing delay at each router. Tulip loss and queuing probings enable the identification of hops where losses occur without actually needing to have explicit control over the hops. This enables an effective way to determine both forward and return path losses on a hop by hop basis along the entire path.

Table 5.1 Some important PlanetLab probing source used and the number of IXP routes visible from each source to top 1000 websites.

Name	Location	ASNum	IXP Routes
chimay-fundp	Belgium	2611	371
cs-surrey-sfu	Canada	11105	270
sos-ac-jp	Japan	2506	118
plab1-sjsu	USA	7132	63
pl2-monash	Australia	7575	407
free-informatik	Germany	680	118
plab2-nec-labs	USA	209	4
plab1-cs-hk	Hong Kong	3363	77
plab-canterbury	New Zealand	9432	185
ops-ii-uam	Spain	766	80

5.1.4 Filtering IXP route destinations from each probing source

Carrying out end-to-end measurements from one source host is evidently not representative of a large section of the Internet. Hence to measure route metrics such as round trip delays

for an entire path, we distribute our probing experiment over a number of carefully selected planetlab vantage points across the world. Due to the differing locations of the various probing sources selected in our experiment, we first carry out traces to the entire list of top 1000 destinations to filter out the respective destinations for which the route traverses an IXP. Different sources have different providers each having their own routing policies which leads to a unique set of destination lists from every source node. Table 5.1 presents relevant information about some of the prominent vantages selected and the number of routes to the top 1000 destinations traversing an IXP.

5.2 Pathneck measurements and results

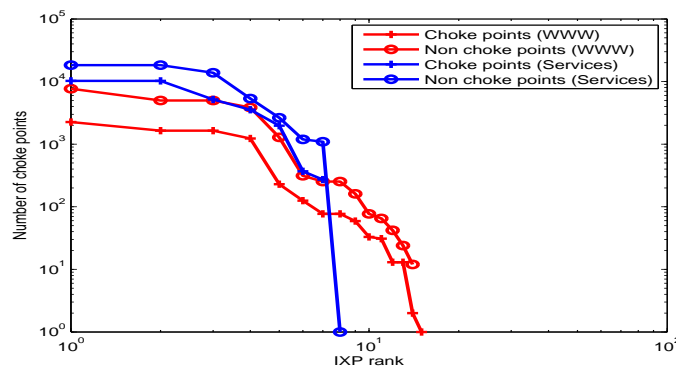


Figure 5.1 Comparing choke point occurrences in IXPs. A limited difference is visible between the number of choke points and non-choke points at IXP hops, denoting a high percentage of choke points occurring at the IXP.

As mentioned in the previous section, the initial set of measurements performed are carried out from one particular host within the UCF network. Traceroutes are conducted to each of the top thousand websites and the paths encountering a known IXP prefix is filtered out. This set of websites using IXPs from the source host is then probed using Pathneck to measure link bandwidth details. We obtain a total of 312 different paths to distinct destinations traversing an IXP. With ten Pathneck probings taking approximately 60 seconds on our host, we cycle through the list of destinations once every 52 minutes. Each Pathneck probe result is the average of n consecutive recursive probe trains (RPTs) with a default value of n being 10. This means a destination is probed once every hour and we carry out this probing for a seven day period. A destination is thus probed 168 times from one host during the course of our measurement.

5.2.1 Identifying choke points at IXP locations

Pathneck returns the top three choke points encountered along the path from the source to the destination. From the measurements we observe that hop locations of these choke points keep changing across the multiple probes being performed. This signifies that the various hops on the path are more loaded than the others at different times of the day. We first filter out all IXP hops along the route and measure the number of choke points occurring within these IXP hops. Interestingly, a high number of IXP hops also exhibit choke point characteristics. Figure 5.1 presents this result. We plot the total number of choke and non-

choke points observed for all IXP hops at each unique IXP location and rank the IXPs based on the number of hops traversed at each. We can observe that the difference in the number of choke-IXP hops and non-choke IXP hops is lesser than a factor of 10 (on a log scale) in the maximum while the difference decreases with other IXPs traversed. The figure presents a useful insight as to how numerous IXP hops are also choke points within the same route. Thus the initial reasoning behind assuming IXPs would essentially allow more efficient and faster data transfer may or may not be true in all cases with the hop taken at the IXP slowing up due to lower available bandwidth. The behavior remains similar for both sets of destination paths, WWW and Services, with paths to the latter having more instances of choke points due to the excess traffic on these links (voice, video or streaming applications).

5.2.2 IXP route persistence

The previous result leads to a requirement to study both queuing delays at IXP locations and end-to-end round trip times to gain a better understanding of the effect of reduced bandwidth at IXP locations. However, route persistence [64] would play a major factor in determining the results of delay based measurements with bottleneck locations along a route varying with a change in the underlying route.

As reported in [62] route persistence may be measured at two levels: *IP level* (also termed as location level) and at the *AS level*. At the IP level, route fluctuations are measured with changes in the IP addresses of every hop along the path. Changes in the IP address of

hops may occur more frequently with different routers within the same AS being selected. Such occurrences lead to a different IP level route but the same AS level route. Only when a more serious network outage occurs would there be an actual variation in the AS level route taken by the packet. Thus, routes at the AS level would show a higher degree of persistence than those at the IP level.

Identifying similar hops at the IP level is a fairly elaborate procedure since multiple IP addresses may be associated with the same router. We follow the heuristics proposed by the authors in [62] to detect IP addresses associated with co-located routers:

- All routers within the source UCF network are considered co-located.
- Routers exhibiting similar location information in their DNS names are considered to be the same.
- After removing digits in the DNS name of the IP address, co-located routers exhibit similar location information.
- IP address prefixes of the IP addresses are also observed to determine if the routers are co-located or not.

Figure 5.2 presents the CDF of IXP route persistence at both the IP and AS levels. The total number of different routes visible from the source to each destination along the IXP route is measured and its CDF plotted. As stated previously, the dataset used to plot this graph uses periodic one-hour probing from the same host for a period of seven days.

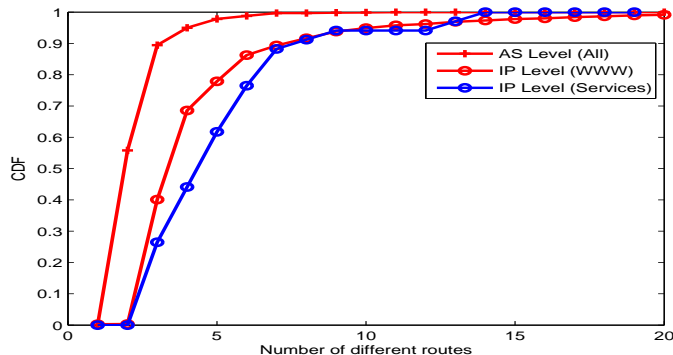


Figure 5.2 CDF of route persistence values of IXP paths for both IP and AS level granularity. The high route persistence values at both levels indicate dominant IXP paths with most paths differing only by 4-5 AS hops over a 7 day period.

We observe greater differences in the number of routes seen at the IP level with close to 85% of these routes differing by a count of five or below. At the AS level¹ the number of different routes visible are noticeably lesser with more than 97% of the routes to the same destination having *dominant* routes with a higher degree of route persistence with less route flapping. The frequency of route changes at the IP level is greater while we observe that no route remains constant throughout the entire probing period.

The observation to be noted while measuring IXP route persistence is the fact that destinations selected are unique with a large volume of traffic being routed to it all times. The requirement for these websites to provide effective service round the clock is of utmost importance which leads to a differing inference than that suggested in [65] where the authors concluded that a third of all Internet routes are short lived. Evidently, most IXP routes

¹We merge all AS level route persistence values in the figure as differences in WWW and Service routes are scarce. This denotes very stable AS level routes for both types of paths.

change to the popular webservers over a period of time but the AS level route persistence shows that this change in routes is not significantly high enough to effect our study of bottleneck links.

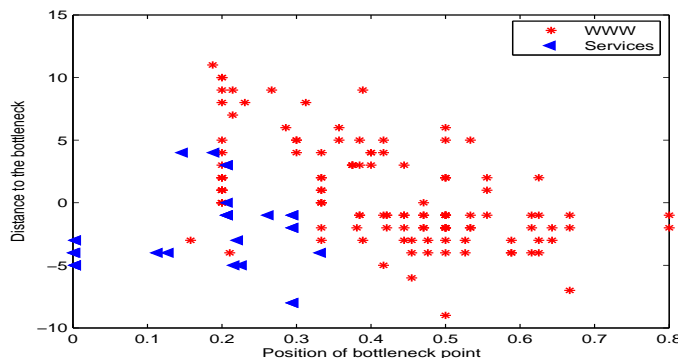


Figure 5.3 Loss position of bottleneck point for IXP hops with respect to the bottleneck position. Most loss points occur within a few hops after the bottleneck hop as shown by a greater number of points with negative distances.

5.2.3 Link losses and queuing delays

Investigating the relationship between IXP bottleneck points and the queuing delays becomes important once bottleneck/choke points have been identified at the IXP hops and the IXP paths have been shown to be fairly persistent. As suggested in [62] the relationship between bottleneck position and loss position helps distinguish between load-determined and capacity-determined bottlenecks. We use Tulip [63] to detect both packet loss and per hop queuing delay from our UCF host to the set of 312 destinations using an IXP route. Tulip

loss and queuing probings are performed one after the other with each probing conducted with 500 measurements (considered the ideal probing set size [63, 62]) for every router along the path. Similar to [62], only forward path loss rates from tulip is considered since Pathneck cannot measure return path loss rates.

Figure 5.3 presents the relation between IXP bottleneck position and loss positions at the IP level. We would like to emphasize here that we only consider those bottleneck hops which are also IXP hops and ignore other bottlenecks observed. Here the x-axis represents the relative position of the bottleneck point with respect to the length of the entire path; i.e. it is the ratio of the bottleneck hop number and the path length. The y-axis represents the relative distance (in terms of hops) of the loss point to the particular IXP bottleneck point. Loss points may occur either before the bottleneck or after it. This is represented by the positive or negative distances respectively, i.e. if the loss point occurs after the bottleneck point then it has a higher hop index and so a negative distance and if it occurs earlier, then its hop index is lesser and so the distance is positive. Loss points occurring at the bottleneck point will have a distance zero.

From figure 5.3 we observe that there is a greater number of loss points before the bottleneck IXP when it is traversed earlier on the route. For example, when the bottleneck position is 0.2 (bottleneck occurring early in the path), the loss points exhibit positive distance. If the IXP hop gets pushed back the loss points start occurring after the IXP hop. In fact loss points generally occur within a few hops after the bottleneck. This would be the regular behavior on any route as hops downstream from the bottleneck would be subject to

a greater traffic load due to the higher queuing affects at the bottleneck hop, which in turn leads to higher losses. Interestingly, we do not observe a high percentage of loss points at the bottleneck point itself (distance equals 0). This would mean that IXPs are not dropping packets but just queuing them for a longer period of time. Figure 5.4 presents the CDF of the distances of loss points observed to the bottleneck. It may be inferred from the figure that almost 50% of the loss points for *WWW* have distances between 0 and -5 and the probability that a distance observed is less than 5 is almost 0.8. For the *Services* dataset almost all the loss points measured are within the first few hops to the bottleneck point (check figure 5.3 to see most services points located very or less than zero distance to bottleneck). When loss points remain close to the bottleneck location it can be inferred that bottleneck IXP location plays a defining role in losses occurring across the IXP route.

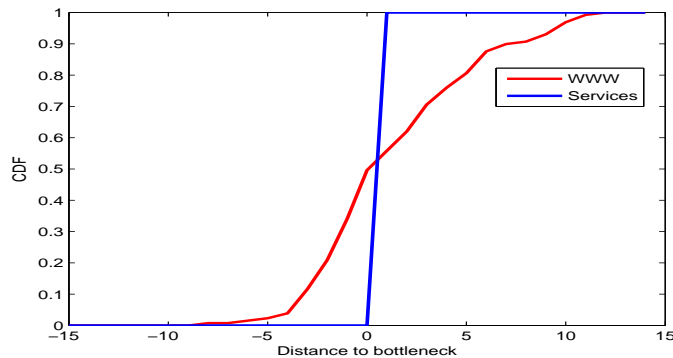


Figure 5.4 CDF of distance from loss point to IXP bottleneck point. Most distances observed are within a few hops of the bottleneck hop which indicate a definite affect of bottleneck choke points at IXP locations leading to link losses closer downstream towards the destination.

Figure 5.5 presents the queuing delays observed for bottleneck and non-bottleneck links in the IXP routes observed. Queuing delay is a measure of congestion (along with packet loss) on a path and is measured by tulip as the difference between the median RTT and minimum RTT from the probing source. Again, the number of 500 measurements for each router along the path provides a reasonable estimate of the queuing delay. From the figure it is very evident that queuing delays at the bottleneck points are much larger than elsewhere with fewer than 97% percent of non-bottleneck links having a queuing delay of less than 5ms, while only about 55% of bottleneck links have a comparable delay value. Overall, queuing delays for a majority of the bottleneck IXP links are much higher than the non-bottleneck IXP links.

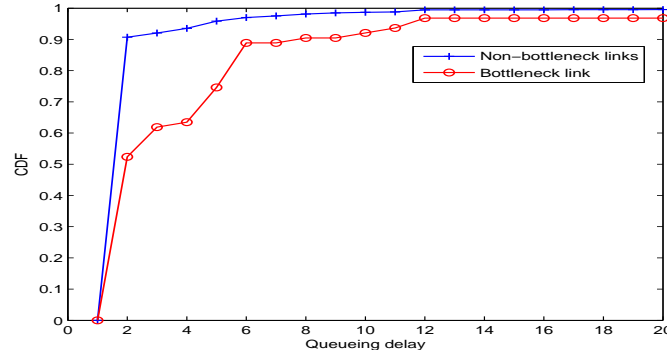


Figure 5.5 CDF of queuing delays at bottleneck IXP links and non-bottleneck links. Queuing delay is calculated from Tulip measurements by computing the difference between median-RTT and minRTT. A higher percentage of bottleneck links at the IXPs are increasing packet queuing latencies.

5.2.4 End to end delay results

We now present the results of experiments on the end-to-end delays between destinations traversing both IXPs and non-IXP hops to the popular top 1000 websites² from the geographically diverse set of PlanetLab vantage points selected. Latencies are monitored by pinging the popular destinations from each vantage point for a period of 24 hours after a traceroute has identified if an IXP path is being taken or not.

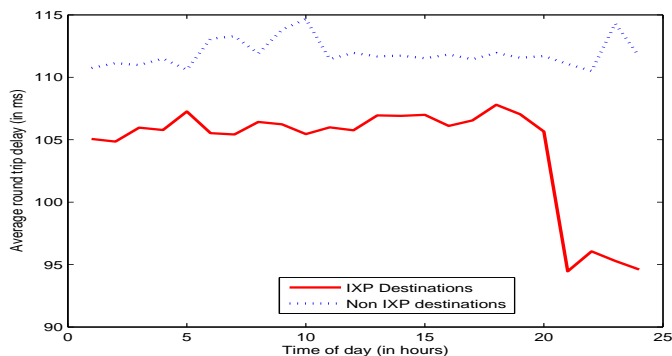


Figure 5.6 Average end-to-end delay obtained from probing sources for IXP and non IXP destinations. IXP destinations have overall lesser latency in comparison to the non IXP paths.

Figures 5.6 presents the comparison between end-to-end round trip delay times observed across IXP and non IXP paths for an entire day of probing. Each of the IXP and non IXP destinations are probed from every source per second for a period of 24 hours. From the previous section, we have observed that route persistence in case of IXPs has been pretty

²we do not select the *Services* destinations in this study. With a variety of destinations using IXP paths from the different PlanetLab vantage points, the measurement process at each location becomes very involved. We restrict our study in this case to only the popular websites.

high and so it would not play a major factor in this time-of-day experiment. We can see that the delay values for IXP routes are constantly lower than those routes not traversing an IXP. We can make a couple of important inferences from this result. Firstly the popular websites have a wide variety of content at different points of presence across various geographical regions each with different providers and hence different Internet routes to these PoPs. As a result the same website may be reached through an IXP path from one location while it may be reached through a non IXP path from another location. For the different paths measured we do see that the IXP paths exhibit lower average latencies than those not traversing an IXP but it has to be noted here that paths from the same source to destination is not being compared. Figure 5.6 is giving an overall view of *all* the paths measured together and not carrying out a hop by hop comparison of IXP and non IXP paths between the same source and destination end hosts. This is an avenue for future work we are currently undertaking to better understand the effect of the IXP by isolating the IXP links along a path. However it is evident that the popular websites would benefit in terms of path latencies if their providers engage more extensively in peering at the exchange points, especially if the websites engage in real time streaming applications/services such as music and media delivery. These applications are sensitive to delay values which in turn effect the Quality of Experience (QoE) of the end users.

CHAPTER 6: TRIANGLE INEQUALITY VIOLATIONS DUE TO IXPs

The basis of design and implementation of Internet overlay networks lies in the identification of detour paths with lower end to end latencies than the default paths. These detour paths lead to an interesting artifact of the Internet delay space when the default direct latency between two arbitrary end-hosts is greater than the computed latency between them via another intermediate host. This characteristic of Internet routing is known as Triangle Inequality Violations (TIVs). In this chapter we study and analyze TIVs created due to peering at exchange points on a global scale from existing measurement infrastructure.

6.1 Internet triangle inequality violations

TIVs in the Internet delay space are a result of different sets of routing policies employed by service providers. These routing policies are designed and adopted based on economic considerations of the corporations with a primary aim of generating profit. While improving routing performance is implicitly linked to these routing policies (better performance will drive more customers to the provider) it is not a primary goal. As a result, the default path across two hosts on the Internet is not always the best route and 30-60% of the time there is an alternate *detour* route with a lower latency [33]. Detour paths with lower latencies

may be formed due to peering as reported by Zheng et al. in [20] which assumes greater significance with the rise in peering across the global Internet [3].

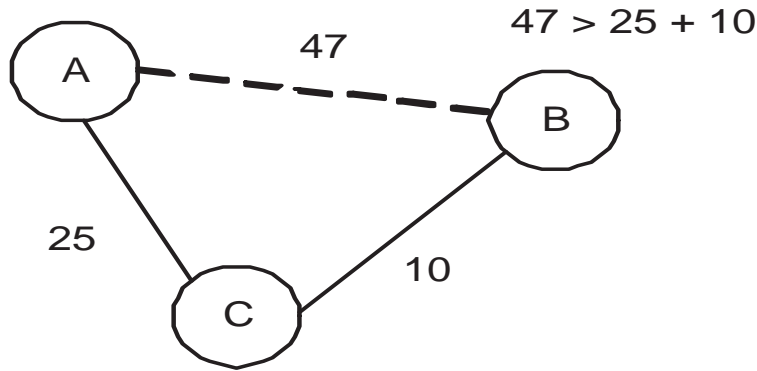


Figure 6.1 A simple example of a triangle inequality violation. Side AB is greater than the sum of AC and CB.

TIVs have been studied and used extensively to implement efficient routing in overlay architectures [22, 34]. As a direct result various applications could benefit if TIVs are leveraged successfully. For example, Ly et al. [23] use detour paths for latency reduction in online games. These detour paths provide a greater diversity of paths which are generally more efficient, have more bandwidth and are often-times much less loaded than the corresponding direct path. These TIVs have also been shown to be widespread [21], with a number of latency data sets collected and used to measure their occurrence and properties [34]. A common characteristic of these data sets is that the actual number of nodes monitored are low, primarily due to the all-to-all nature of these measurements. Latency measurements are made from every node to all other nodes in the set, or at-least a high percentage of the remaining nodes ([34] uses 200 and 1715 nodes, one data set in [21] uses 4000 nodes, [22]

uses 1953). This technique does not lend itself to scale easily thereby making a study of TIV characteristics restricted.

While the above mentioned data sets have been used widely and has enabled the determination of important TIV characteristics, we take a different approach in this work. Our primary aim is to study how increased peering in the global Internet is affecting the formation of TIVs and if it could be leveraged successfully. IXPs have now become a popular fabric of the Internet ecosystem and obtaining latency information of paths traversing IXPs (henceforth called IXP paths) required us to consider the use of a measurement architecture on a large, global scale. CAIDA's Ark [66] measurement infrastructure provides the ideal data; traceroute information enabling the easy identification of IXP paths while latency measurements provide round trip times (RTTs) along the entire path. With 54 source monitors located worldwide sending probes to every /24 network on the Internet, we obtain a data set with millions of valid paths available thereby enabling the study of TIV characteristics on a very large scale.

We now study the characteristics of TIVs created **only** due to ASes peering at IXP locations [67]. Lumezanu et al. in [34] propose the concept of mutual advantage between hosts that benefit from each other, an idea which is quite similarly put into use by ASes mutually agreeing to peer at an IXP. We measure the effects of these specific TIVs on inter-domain routing and try to determine if they could be useful in designing routing overlays. We then carry out a detailed graph based study of the detour paths forming TIVs and identify

particular characteristics, specially a set of IXP links popular in the detour paths. These links contribute to the formation of TIVs and improve the end to end latency.

6.2 Experiment setup

6.2.1 Dataset selection

We describe our methodology behind measuring large scale end-to-end TIVs between networks across the global Internet. With our focus on obtaining routes comprising of a peering link at an exchange point, we have two essential requirements to be satisfied in our experimental setup:

Latencies:Source to destination end-to-end latencies to search and compute TIV occurrences.

IXP hops: Intermediate route hop interface IPs to identify the presence of an IXP along a route. We use a set of known IXP prefixes from PCH [1], PeeringDB [15] and the IXP Mapping project [16]. Note that a route will almost always have at most one IXP hop due to the valley free property of Internet routes.

CAIDA’s Archipelago (Ark) active measurement infrastructure provides large scale traceroute-based topology measurements to all routed /24’s (9.1 million) simultaneously with a team based parallel probing mechanism. With individual source to destination latencies

along with a complete hop-by-hop trace available, we find their available dataset the most suitable for this study.

The TIV study is carried out in two phases: we obtain the first complete cycle of Ark traceroutes from all 54 CAIDA monitors (divided into 3 teams) in October 2010 (a probing cycle typically completes in 2-3 days) which we call **ARKOct**. Our study of TIV characteristics is based on this dataset. Our follow up graph based TIV study is over an extended time period (we discuss the reasoning behind this in a later section) for which we identify TIVs for an entire month of Ark traceroutes in January 2011. A total of 32 cycles (with $\tilde{288}$ million paths) of traceroutes are analyzed to identify TIVs due to exchange points and the underlying AS paths are uncovered (this dataset is called **ARKJan**). Table 4.3 summarizes the salient features of the datasets used in our study and we release our data sets and results at [68].

6.2.2 Identifying end to end TIVs due to IXPs

The first step in identifying TIVs is to list all direct paths available in the traceroute dataset and filter out the IXP-paths. For a given cycle of data, we first parse out all the *src-dest* IP pairs and their end to end latencies (giving the triple $(srcIP, destIP, RTT)$). All IXP-paths in the cycle are identified using the available IXP prefix lists leading to the $(ixpSrcIP, ixpDestIP, ixpRTT)$ triple list. The IP addresses on these lists are now mapped to the corresponding AS numbers to end up with an AS level map of $(srcAS, destAS, RTT)$

triples for both IXP-paths and other regular paths¹. Multiple $(srcAS, destAS)$ combinations are possible for the latter list as different monitors could probe different IP addresses within the same destination AS. We hence simplify our latency calculation by computing the mean RTT for all instances of a given $(srcAS, destAS)$ pair. An obvious downside of computing mean RTT values has been discussed by Lumezanu et. al. in [70] where the authors suggest median values may create the illusion of TIVs. However, in our analyzed data we observe a high number of $(srcAS, destAS, RTT)$ triples with very similar end to end RTT values. Computing the mean RTT for these paths provides a more representative idea of the actual RTT between the respective source-destination ASes and also reduces duplicate records of detour paths between these ASes.

From the previous step we have obtained the set of direct paths between every CAIDA source monitor and the destination ASes across the global Internet. We also filtered those direct paths using an IXP. Finding the detour routes traversing an exchange point (through another source) for every direct path is the next step. Since inter-monitor latencies are not mentioned in the CAIDA data, we calculate an approximation of the RTT between each monitor by probing all 54 monitors from one host (within our campus network) and then computing the relative latency between the monitors with respect to our host. Even though this value is not accurately representative of the RTT between the source monitors, it is the best alternative available. Also, TIVs mostly occur when the latency is significantly lower in a detour path between the $(srcAS, destAS)$ pair and not the sources. As a result, a relative

¹IP to AS mapping is a difficult and inexact problem in its own right, but for consistency here we use the Team Cymru whois service available at [69].

approximation of the latencies between sources does not affect the creation or destruction of TIVs in the measurement system.

We now choose every default direct path (both IXP and non-IXP paths) and compute detours through another source and an alternate IXP path with reduced latencies. We call these detour paths *IXP detours* and unless otherwise mentioned all detour paths in the rest of the paper are these IXP detours. These IXP detour edges are computed using the Floyd-Warshall shortest path algorithm; which is both memory efficient when used with large graphs and enables path reconstruction to help obtain the intermediate hops along the shortest path. We select the best available shortest IXP detour and record the original latency, the new shortest latency and the detour path.

Algorithm 1 Identifying global TIVs

Require: *cycle*: Traceroute based scamper data cycles from CAIDA,

ixpPrefix: List of IXP prefixes

- 1: Filter (*srcIP, destIP, RTT*) for every default path to *allList*
 - 2: Find (*ixpSrcIP, ixpDestIP, RTT*) using *ixpPrefix* to *ixpList*
 - 3: Map *allList* and *ixpList* to (*srcAS, destAS, RTT*) using [69] (name *allASList* and *ixpASList*)
 - 4: Compute mean RTTs for duplicate (*srcAS, destAS*) pairs in *allASList*
 - 5: Compute inter-source latencies and store in *srcASList*
 - 6: **for** every path in *allASList* **do**
 - 7: Concatenate *srcASList, ixpASList* to path
 - 8: Run Floyd-Warshall algorithm and select best alternate detour as IXP detour
 - 9: Record *srcAS, destAS, RTT*, IXP detour RTT and path
 - 10: **end for**
 - 11: Repeat for next *cycle*
-

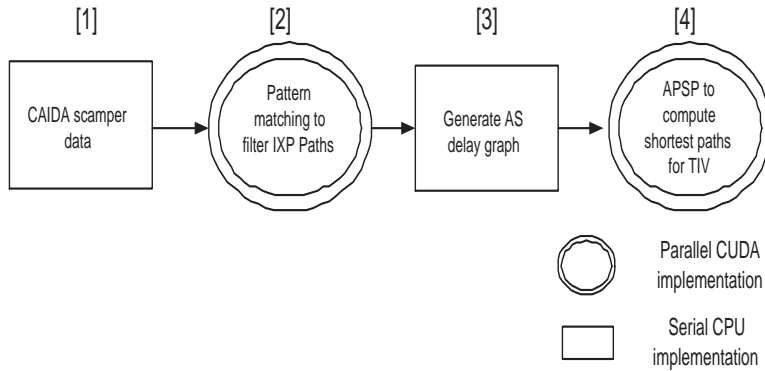


Figure 6.2 The sequential and parallel modules in the TIV identification process. The pair of parallel modules (2 and 4) are implemented in CUDA.

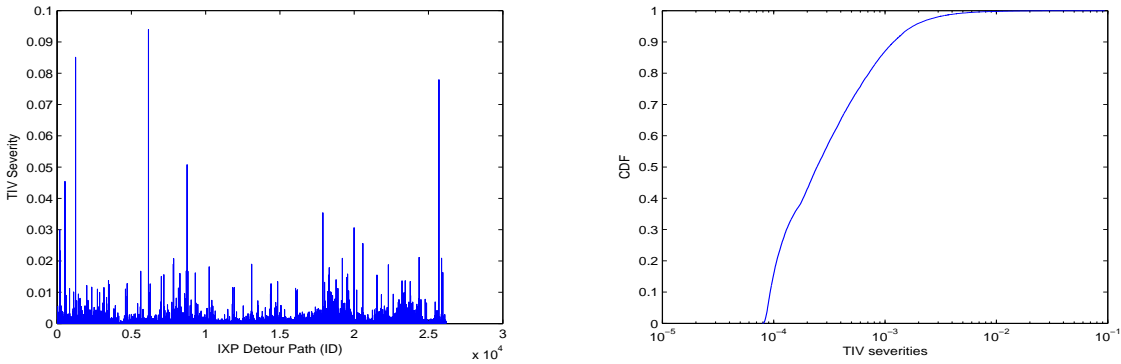
6.2.3 Modules and parallel implementation

- **Read the CAIDA data:** The data files are generated by CAIDA’s scamper probing tool and contain thousands of routes with individual hop information. These files are large, typically 4GB per cycle which must hence be read by our parser in parts.
- **Pattern matching to filter IXP path:** Searching and identifying IXP routes from larger traceroute based datasets presents our first opportunity to invoke parallel processing by applying *pattern matching*. With millions of routes to search from for a specific number of IXP prefixes (we use the list of 373 prefixes from [16]) this important step in the process of identifying TIVs can be efficiently implemented in a parallel fashion.
- **Generate AS delay graph:** The process of doing the IP-to-AS conversion of the source, destination IP addresses are now carried out in a serial fashion. Querying the

team cymru service for the conversion and then replacing the IP addresses with their respective ASes is done serially.

- **All pairs shortest path (APSP):** With the IXP-paths filtered out earlier the result of this step gives us the shortest detour path through an intermediate source and going through atleast one IXP hop along the way. APSP is another well-known candidate for parallel GPU programming and we use it to reduce our overall TIV route identification time. We discuss the APSP implementation in the next section.

6.3 TIV Characteristics



(a) TIV Severity variations

(b) CDF of TIV severities

Figure 6.3 Observed TIV severity information. A large range of severities are observed and the low absolute values are due to the high number of nodes in the delay space.

We study the characteristics of the end to end TIVs obtained from the *ArkOct* dataset in this section. Wang et.al. in [21] carried out a detailed analysis of TIV from four different

Internet based datasets with an evaluation metric termed *TIV Severity*. They define the TIV severity of edge AC of two nodes $A, C \in S$ as:

$$Sev = \frac{\sum d(A, C)/(d(A, B) + d(B, C))}{|S|} \quad (6.1)$$

where S is the delay space with all nodes, $B \in S$ and $d(A, C) > d(A, B) + d(B, C)$

In computing the TIV severities, we consider only the significant TIVs [34]: where the detour path reduces the direct path latency by atleast 10ms and 10%. Figure 6.3 presents the observed severity values in TIVs formed due to IXP detour paths. Severities range from 0.0001 to 0.1 with close to 80% of TIVs exhibiting a severity of 0.001 or below (from fig. 6.3(b)). This means that more severe violations are caused only by a few edges in the detour paths (fig. 6.3(a) shows the differing severity levels observed) and the distribution is heavy-tailed.

A major difference between the observed TIV severities reported in [21] is the near order of magnitude difference in severity values seen, an interesting characteristic. The datasets used by the authors in [21] were smaller, the maximum number of nodes was 4k while in *ArkOct* we see 12722 unique nodes. Due to severity levels being normalized by the number of nodes in the delay space ($|S|$) we may naturally infer lower severity values. However, with a threefold increase in the number of nodes, we see almost a three orders of magnitude decrease in the TIV severity levels. This means IXP detours are not the only source of significant TIVs. Regular non-IXP detour routes also reduce path latencies. The

increased peering in the modern Internet ecosystem [2] is thus not leading to large TIVs even though the underlying topology is incurring a sea-change.

6.3.0.1 TIV severity with delay

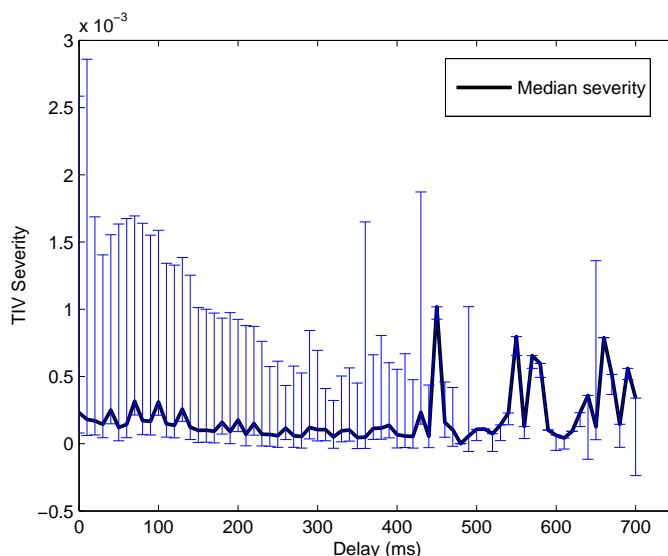


Figure 6.4 Median delays and severities for TIVS generated due to peering.

Figure 6.4 provides an indication of how detour edge delays affect the severity of the TIVs they create. Similar to [21] we separate all TIV edges into 10 millisecond bins and plot the corresponding severity of TIVs within each bin. The median severity remains similar for delays upto 400ms and only for the longer edges do these severities start changing. This means TIV severity for IXP detours vary for longer latencies. However the 90th percentile error bars show that the TIV values for a particular edge length do fluctuate for smaller routes ($< 300\text{ms}$) leading to the conclusion that smaller routes are susceptible to more severe TIVs

instead of only the longer routes. IXP paths overall are not free of the TIV phenomenon and the view put forward in [20] of peering leading to TIV formation is certainly true.

6.3.0.2 Detour hop characteristics

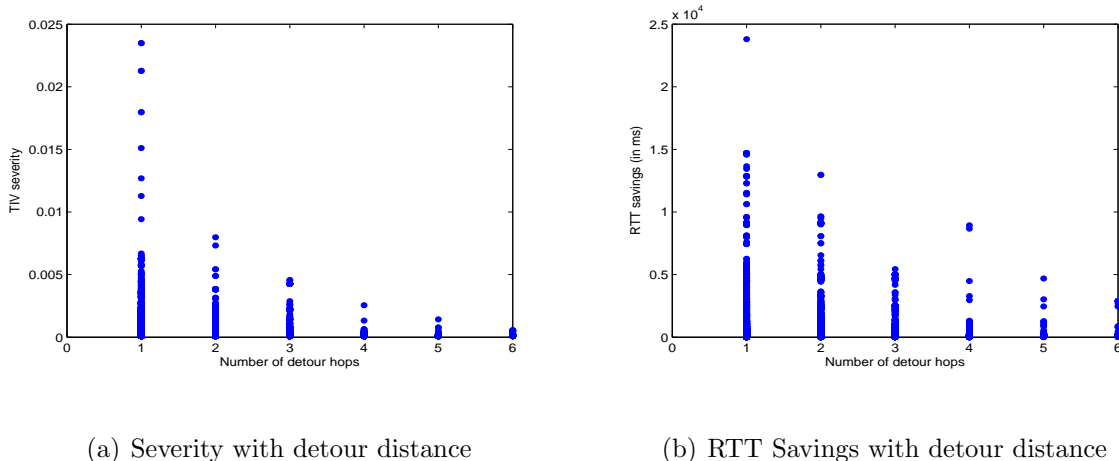


Figure 6.5 Detour path characteristics. One intermediate hop show greater severities and larger latency savings.

We now study hop characteristics of all the IXP detour paths observed in the Ark measurements. In our alternate path creation strategy, we do not limit the number of detour hops to a single alternate source to ensure the computation of all available detours. From the set of available detours we select only the *best* detour path (with the greatest RTT savings). Figure 3(a) presents the observed TIV severity values of all the detour paths against the number of detour hops. We observe the highest number of optimal detour paths have a single intermediate hop (as also shown in [22]) and the greatest variation in severities. The

severity level decreases with the increase in number of hops taken by the detour. Longer detours hence do not lead to more severe TIVs which reinforces the proposition that an RTT improvement will most likely be made by detouring through *one* intermediate node. Figure 6.5(b) strengthens this observation. RTT savings are highest for one hop detours (which also lead to more severe TIVs) and gradually decrease with increasing detour path length. Note that a greater RTT saving does not always equate to a higher TIV severity. This is because the severity of an edge is larger when it causes a higher number of violations. Every violation on the other hand could be either large or small generating a latency reduction of any magnitude.

6.3.1 Detour path graph properties

We now study TIVs from a graph based perspective with the aim of trying to identify trends in the background AS-AS linkage suggestive of these violations. We first discuss the graph creation procedure and then present details about the macroscopic properties of these TIV graphs.

6.3.1.1 Graph creation

A widely used technique in Internet topology evolution is ([26, 10]) to generate AS-level maps of the Internet by combining visible AS-AS links from different datasets over a period

of time (typically a month or more) and create a snapshot for the chosen period. The focus is primarily on coming up with a representative graph of the Internet. We employ a similar but simpler approach in creating a graph comprising of the detour edges which generate the TIVs.

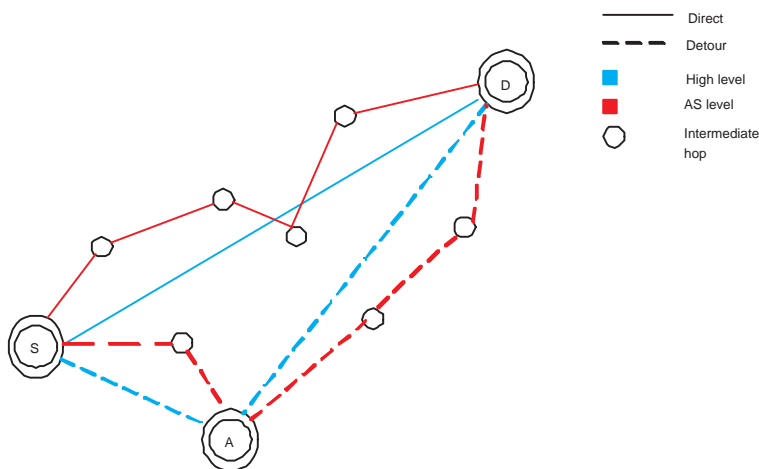


Figure 6.6 Simplified example of high level and AS-level edges. The solid lines denote direct paths while dotted lines are detours. Blue denotes a high level edge while red shows the AS-level edges from which the respective graphs are constructed.

We identify TIVs based on the procedure specified in the previous section for entire cycles of Ark data in the course of a month (Jan 2011). We run our parsing scripts and carry out TIV identification for probing cycles for every team. A probing cycle typically lasts for 2 days per team but the simultaneous probing architecture of Ark generates 32 total cycles divided among the 54 Ark monitors. We then obtain all the direct paths and the detour paths through intermediate sources for each cycle and separate the detour paths forming significant TIVs. Every hop in the detour path is now a link from which the detour graph

is constructed (by combining all the computed IXP detours). It comprises of the source AS, intermediate sources for the detour and finally the destination ASes. Paths are then broken down to individual edges and duplicates removed resulting in the graph of all detour links forming TIVs for the entire month.

We then divide our graph study by creating two detour graphs:

High level: We construct a graph comprising of the edges connecting the source-destination ASes through the intermediate source and disregarding the individual lower level inter-AS links. We name this as our high-level view (G_h). The procedure described above gives us this detour graph.

AS Level: To study further the properties of detour paths, we reconstruct the high-level detour edges with the internal inter-AS links along the source to the detour hop onwards to the destination (the graph is called G_a).

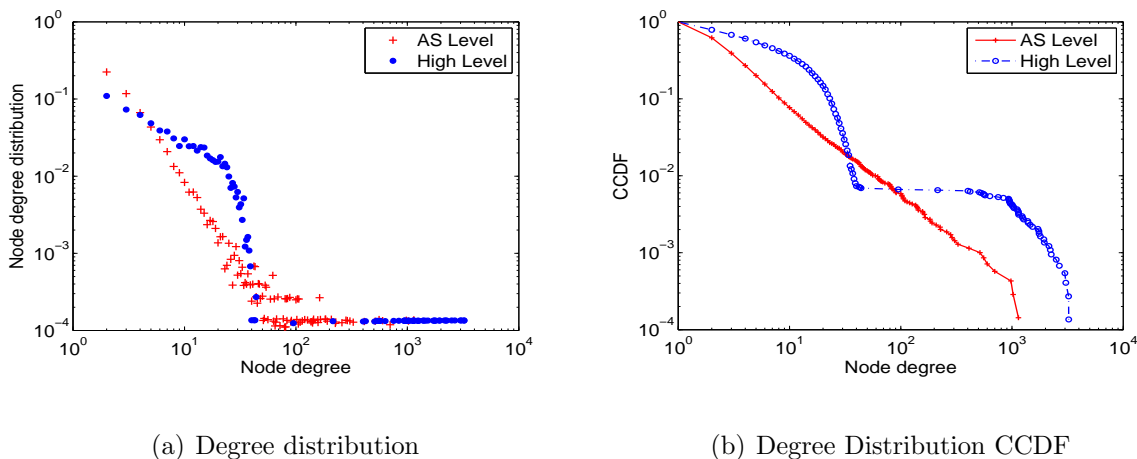


Figure 6.7 Node degree distributions of the High-level and AS-level graphs.

Figure 6.6 shows the direct and detour paths from a source (S) to destination (D) via hop A along with the high level (blue) and AS level (red) links.

6.3.2 Graph characteristics

We first compare both graphs in terms of node/links to get an idea of the basic structure of the graphs. Table 6.1 presents this information. One would expect more nodes and links to be visible at the AS-level since it presents hop by hop information along the detour paths; but Table 6.1 shows otherwise. There are close to 400 more unique nodes with almost 47K extra links in G_h compared to G_a . The low number of links in G_a (19K vs 65K in G_h) presents us with interesting information about the nature of detours through IXPs. This means most IXP detours use the same underlying links which lead to creation of the TIV.

Table 6.1 Comparing the number of observed links in the high level (G_h) and AS level graphs (G_a)

	Nodes	Links
G_h only	1005	63409
G_a only	616	17402
Both ($G_h \cap G_a$)	6366	1776

Entirely different direct paths have a common subset of better detour links through some common IXPs. These links (which are relatively low in number) provide lower-latency paths serving as popular detours to the high number of default direct paths.

6.3.3 Degree distribution

The node degree distribution is the most widely used connectivity metric specially since Faloutsos et al. [45] proposed the degree distribution of the router level Internet to follow power laws. With these types of distribution being highly variable and traceroute based studies susceptible to different biases, later work such as [50] have comprehensively demystified the *scale-free Internet myth*. However, since we do not study complete Internet topologies here, this does not have any direct consequence on our analysis. It provides useful information about graph connectivity.

Definition: Degree distribution is the probability that a node selected randomly is of k -degree and is calculated by: $P(k) = n(k)/n$ where $n(k)$ is the number of k -degree nodes in a graph with n nodes.

Discussion: Figure 6.7 presents the degree distributions of both graphs obtained. A lower number of low to medium degree nodes in G_a leads to lesser probability values for these type of nodes (in 6(a)). Probabilities increase for the medium degree nodes (around degree 100) denoting high connectivity in the core of the graph, a property not exhibited by G_h . This reinforces our earlier idea that a high percentage of nodes in the AS level graph

are well connected with significantly lesser number of low degree nodes than in G_h (as shown in fig 6.7(b)). Greater node connectivity would invariably lead to higher edge utilization, a property we study next in terms of betweenness.

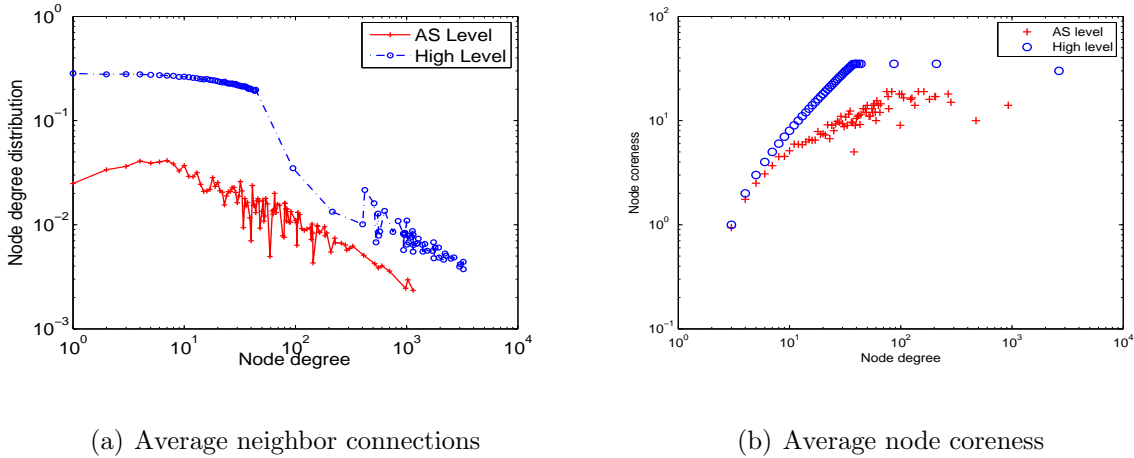


Figure 6.8 Average neighbor connections and node coreness.

6.3.4 Joint degree distribution

The joint degree distribution (JDD) gives us an idea of the general *neighborhood* of a randomly chosen node with average degree. The immediate one hop neighborhood of the node gives significant information about the structure of the area around the node.

Definition: The JDD is the probability that a randomly selected edge connects k_1 and k_2 degree nodes: $P(k_1, k_2) = \frac{m(k_1, k_2)}{m}$, where $m(k_1, k_2)$ is the total number of edges connecting nodes of degree k_1 and k_2 . The average neighbor connectivity (a summary statistic of JDD) [26] is the average neighbor degree of the k degree node. The maximum value for the

neighbor connectivity in a complete graph is $n - 1$, which is what we normalize with before studying this metric in figure 6.9(a).

Discussion: Figure 6.9(a) shows that G_h has a high number of radial links connecting high degree nodes to low degree nodes; namely the few source nodes to numerous destination nodes respectively. This results in a high neighbor connectivity value since a destination node is always directly connected to the source, which is an inherent limitation of our data (we discuss limitations in a later section). In the AS-level graph the radial links uncovered again point out the popularity of IXP nodes. Peering ASes connect at these nodes raising their node degree but are themselves connected to fewer number of nodes which are customer/provider ASes.

6.3.5 Average node coreness

The node coreness provides more information regarding node connectivity as it exhibits how well the node is connected to the entire graph and not only its neighbors. A node may have a high degree but its connectivity to other parts of the graph is dependent largely on its neighbors. The best example to describe this is a high degree hub of a star; it has a coreness of 0 if its neighbors all have a degree 1. When these neighbors are removed the hub is left disconnected.

Definition: The k -core of a graph can be defined as the subgraph obtained from the original graph by the iterative removal of all nodes of degree less than or equal to k [26]. The

node coreness (κ) is the highest k for which the node is present in the k -core but removed in the $(k + 1)$ -core. Thus all one degree nodes have coreness equal to 0 while the maximum node coreness κ_{max} is termed the *graph coreness*.

Discussion: Figure 6.8(b) plots the averaged node coreness of the two graphs with increasing node degrees. The results reinforce the conclusions from the analysis of the average neighbor connectivity in the previous section. The high level graph has greater node coreness since due to the high node connectivity of the sources while at the AS level the IXP nodes (with greater degrees) are connected to the lower degree ASes peering at those locations. These IXP ASes are *deeper* in the core of the graph but not extremely well connected. The importance of these nodes and their links are measured in the next section by the betweenness centrality metric.

6.3.6 Betweenness

Betweenness is a widely used measure of centrality applicable to both nodes and links. Nodes which appear on a greater number of shortest paths between an arbitrary pair of nodes in the graph exhibit a higher betweenness value. Such nodes are considered to be more *central* than others since a high percentage of traffic would be routed through these nodes based on the assumption that the traffic is uniformly distributed across all nodes and links.

Definition: If σ_{ij} is the number of shortest paths between nodes i and j and l is either a node or link; then $B_l = \sum_{ij} \sigma_{ij}(l)/\sigma_{ij}$ is the betweenness of l [26]. The value is

normalized by the number of pairs of nodes not including itself, which is $(n - 1)(n - 2)/2$ for undirected graphs.

Discussion: Figure 6.9(b) presents normalized node betweenness values for both high level and AS level graphs. As expected from the earlier discussions inferring the popularity of the AS level links, we observe higher betweenness values for nodes across the entire range of degrees (low to high). This means most nodes in the AS level graph are popular and heavily used in the paths from the sources to all the destination ASes. In the high level graph even though the detour paths traverse an IXP in all cases, betweenness values are generally on the lower side; which means most of the high level shortest paths are disjoint. Popular detour nodes (which are other source nodes in this case) are lesser than those at the AS level. These lower level popular ASes are the IXPs or the customer ASes peering at the IXP switches.

Edge betweenness is however a more complicated metric which provides different insight into the structure of the graph studied. The authors in [26] propose it to be a measure of a combination of link centrality and radially in a graph. We compute the edge betweenness centrality as a function of end node degrees (shown in figures 7(c,d)). The high level graph shows lowest betweenness values are between tangential links connecting low and high degree edges even though we would expect the betweenness of links connecting low degree nodes to be the least. The observed behavior is an artifact of our detour computation procedure where the sources (which are very few in comparison to destinations) are connected to a high number of end hosts at the graph fringes. This is not the case in the AS level graph

where tangential and radial nodes within all degree ranges display very low edge betweenness (as shown in fig 6.9(d)). Higher edge betweenness in both graphs are evident in edges with greater node degrees but the presence of greater number of high range tangential edges in the high level graph generates more edges with higher edge betweenness values. The AS level graph contains more radial links leading to the high concentration of edges with low edge betweenness values.

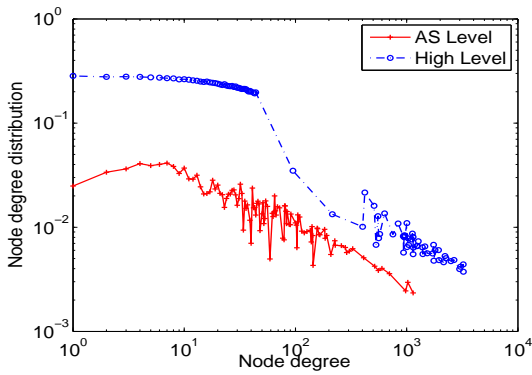
6.4 Discussions and conclusions

The CAIDA Ark dataset presented few limitations due to the nature of the measurement process. Firstly, the sources sending out the probes to all destinations is limited to 54, not a particularly high number. These source monitors are scattered worldwide but all detours are routed only through these nodes making our TIV computations dependent on them. Other latency data sets used to compute TIVs sometimes have *All-to-All* information for a richer set of TIVs. However the cost of such an approach is that the data set increases exponentially with an increase in the number of nodes and hence becomes impractical for large node sets. This characteristic led us to explore the parallel implementation of TIV identification using the readily available NVIDIA GPUs. The primary goal of this work is to observe creation of TIVs on a global scale, for which the Ark data is a very suitable fit.

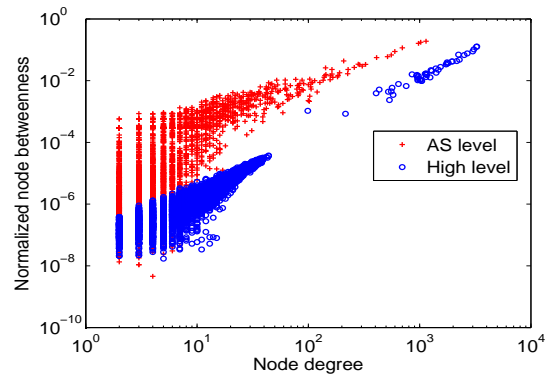
We first identify steps in the original serial implementation which are potential candidates for parallelization and separate them out as distinct modules. These modules are then

taken individually and implemented in a parallel fashion with CUDA after we determine the exact type of operation being carried out. The IXP route identification step is essentially a pattern matching exercise while All-Pairs Shortest Path scheme is required to compute the shortest paths between two nodes in a graph. Both these problems have been the subject of great attention in the parallel computing community from which we study and obtain the proposed solutions before implementing them in relation to our problem at hand. We combine the results of these different parallel modules alongwith other stages in the entire process which are either not parallelizable or would present little or no benefit overall. This parallel approach to solving our problem yields hugely significant gains in performance and efficiency in the raw running times in comparison with reference serial implementations.

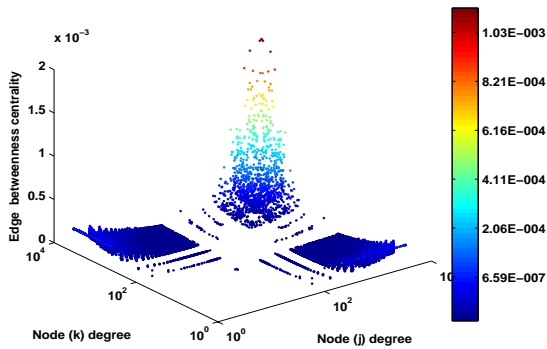
Once the global TIVs are identified we analyze them in detail to further our understanding of their properties. An established severity analysis of these TIVs due to IXPs exhibit characteristics different from traditional TIVs mainly due to the global nature of the data and leads us to investigate more deeply into the underlying frameworks. We carry out a graph based study of these IXP TIVs to provide an initial theoretical basis of indicating the popularity of peering links responsible for most detours.



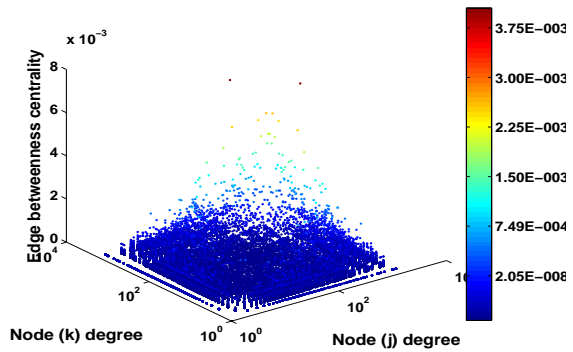
(a) Average neighbor connections



(b) Node betweenness



(c) High level edge betweenness



(d) AS level edge betweenness

Figure 6.9 Graph properties of AS level and High level graphs. The betweenness graphs (b-d) denote the popularity of the IXP links at the AS level resulting in higher node betweenness values. In (c)&(d) the x,y axes denote node degrees of two arbitrary nodes j&k.

CHAPTER 7: PARALLEL FRAMEWORKS FOR ANALYSIS WITH GPU

The computation of TIVs across millions of Internet routes at the AS level is characteristically a candidate for distributed processing. Also the sheer magnitude of the job lends itself well to an efficient parallel processing solution. We describe and detail the various steps in the process of TIV identification (in the following sections) and group the major steps into independent modules, each of which can be associated with different disciplines of parallel computation. We apply parallel programming techniques to these modules to improve and adapt our overall approach towards solving the problem of TIV computation. We then implement efficient schemes available in distributed computing literature to parallelize the entire process first on a multi-core CPU and follow it up with an implementation on the NVIDIA GPUs using the CUDA interface [71]. The presence of thousands of threads readily available on the GPU greatly enhances the efficiency of our original algorithm and ultimately results in significant savings in overall compute time. We observe that a serial approach typically takes more than an entire day in identifying global TIVs, while the parallel scheme reduces the time taken to a few hours, on the same dataset. We report our timing measurement results and discuss paths for future improvements in processing time and memory efficiency.

Identifying TIVs on a global scale is the first step in studying their characteristics. In this work we concentrate on identifying the aspects of the TIV identification process which

are parallelizable and implement some well known parallel solutions to these steps in the process [72]. We run a wide variety of experiments on various platforms and hardware and observe significant savings in the time taken for analyzing a complete dataset. More than 280 million Internet paths are analyzed and IXP paths identified from which TIVs are computed. The main algorithm is divided into four modules of which two are prime candidates to be implemented in parallel. These two modules were implemented in parallel using the Matlab Distributed Computing Toolbox (DCT) and CUDA on multiple CPU cores and the GPU respectively. Individual speedups are recorded before the entire algorithm is studied in detail from which we obtain a speedup of about 4x in a parallel CPU implementation. A 2x speedup from the parallel implementation times is observed in the entire process when the GPU is used. In summary our main contributions are as follows:

- We compute global TIVs (see fig 7.1) on a large scale with paths only through IXPs and find a large number of detours comprising these TIVs. Ours is the first such study concentrating on using parallel techniques to measure and identify TIVs due to peering/IXPs and detour routing across the Internet.
- We identify steps in the TIV identification process which are amenable to parallel solutions and carry out these implementations in detail. We experiment exhaustively to obtain timing improvements on a variety of platforms with both CPU and GPUs.

- We observe 4-8x speedups on the entire process while the individual steps implemented in parallel exhibit higher speedup values upto 35x. This shows the effectiveness of the parallel solutions and potential in using these for large scale network data analysis.
- We release our computed datasets and further results at [68] for further analysis and use by the community.

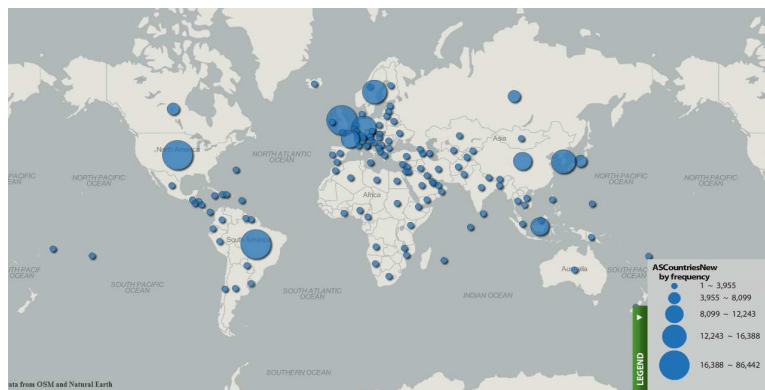


Figure 7.1 AS locations of computed TIVs in our study. Countries in North America and Europe naturally have the highest number of ASes visible since a majority of routes measured in the datasets are hosted in these locations.

7.1 Platforms

In this work we carry out parallel implementations of the serial algorithm in two different platforms. The ready availability of multi-core architectures and software tools such as Matlab DCT enables us to implement a parallel version of both pattern matching and APSP

algorithms. Parallel implementations which are executed on the CPU provide an appropriate benchmark to measure timing improvements obtained by the GPU implementation.

Our primary CUDA enabled device is a Dell Alienware X11 computer with an Intel Core 2 Duo CPU running at 2.80GHz and 4GB of memory. This machine has 2 CUDA capable devices, the Nvidia GeForce GTX 260M with 112 CUDA cores, 1GB global memory and the GeForce 9400 with 16 CUDA cores and 256 MB of memory. Both GPUs have 8192 registers available per block and clock speeds of 1.35 GHz and 1.10 GHz respectively. We run the CUDA driver version 4.0 on the same machine with the GTX 260M as the primary GPU. For comparison we first implement an optimized serial code and parallel code in Matlab using its Distributed Computing Toolbox. The CPU experiments are executed in the following settings:

1. Serial CPU code on Intel Core 2 Duo T9600 @2.80GHz with 4GB RAM with 6MB shared cache running either Windows 7 or Fedora 13 with dual-boot. Serial code for pattern matching is carried out using Linux utilities in Fedora. The APSP implementation uses Microsoft Visual Studio 2010 with C++ and we refrain from using any built-in libraries (such as Boost).
2. The serial code for APSP is also run on a desktop with Intel Core 2 Quad Q6600 @2.4GHz with 4GB RAM and 8MB shared L2 cache running Fedora 13 (Goddard) and the linux kernel 2.6.33. This platform provides a perspective on the run-times for APSP running on both Linux and Windows machines. The timing results are presented in a later section.

3. We implement a parallel version of the code in Matlab which uses the Distributed Computing Toolbox (DCT) to run it in parallel. We install Matlab on the same computer as in [1] above with the distributed computing server located on the same machine. This eliminates network communication delays between the Matlab workers and the server and the only overhead generated is due to the parallel implementation of the algorithm.

Parallel CPU implementations of pattern matching and APSP both exhibit faster run times in comparison to the serial CPU implementations. These parallel times enable us to gain a better understanding of the speedups obtained in the parallel versions of the algorithm. The following sections presents both the pattern matching and APSP algorithms in detail discussing the parallel CPU and GPU implementations of both.

7.2 Pattern matching in parallel

This section presents the timing results for both the parallel CPU implementation of the pattern matching scheme and the GPU implementation of PFAC in CUDA. The parallel CPU implementation in Matlab is presented first. We then present a more detailed analysis of the better performing GPU implementation with PFAC.

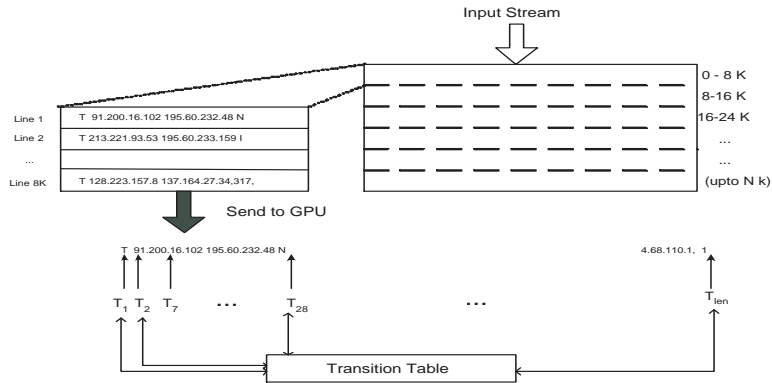


Figure 7.2 Pattern matching with PFAC in CUDA. A new thread T_i is started at every character per line from the input stream and matched using the Aho-Corasick technique with the transition table of IXP prefixes. When a match is successful, all other threads are stopped and the entire line returned along with the match position.

7.2.1 Parallel CPU implementation and timing results

We first implement a parallel version of the pattern matching scheme in Matlab using the DCT. The platform provides quick and efficient techniques to run parallel code with the in-built toolbox handling the inter-process communication details. Table 7.6 presents the times obtained for the parallel CPU pattern matching implementation we carry out in Matlab. In comparison to the serial version, we observe speedups in the range of 1-2x for all the dataset cycles. The Matlab DCT reduces communication but by the very nature of our pattern matching scheme, we can only search one route for the set of IXP prefixes at a time. This results in significant times for the traceroutes to be loaded into memory only a few of them at a time thereby limiting the potential for timing improvements. However the speedups

obtained translate to a couple hours of execution times, which do represent an improvement on the overall times.

7.2.2 GPU implementation details

Searching for a certain set of IXP prefixes on the GPU is carried out using the Parallel Failureless-AC algorithm (PFAC) [73]). The authors in [73] propose a variant of the popular Aho-Corasick (AC) [38] algorithm exploiting its parallel features. PFAC creates an individual thread for every byte in the input string from which the pattern matching is carried out. Using a pre-created state machine of the patterns to be matched, a thread starts the matching at every position on the input stream. Whenever a thread does not find a pattern (reaches a failure state) it terminates immediately with no requirement for a back-track transition to a failure state. As a result, in PFAC there are no failure transitions in the AC state machine with each final state representing a unique pattern. Such an approach creates a high number of threads (an average input stream for a traceroute has about 1500 characters) but most threads terminate very early due to matching failures. These threads are created at successive memory locations on the input stream providing better spatial locality while enhancing usage of the high speed shared memory available to the GPU. Moreover, CUDA's memory coalescing property allows the GPU to combine memory accesses to consecutive DRAM locations in a consolidated fashion as one single read request, thereby delivering near optimal bandwidth on global memory. These enhancements make PFAC a fast, ideal

solution to our problem of identifying IXP prefix strings on an input string with a set of IP hops and RTT times. Figure 7.2 details the application of PFAC to the traceroute datasets used in this work.

7.2.3 CUDA implementation timing results

Figure 7.3 presents a direct comparison between times taken to carry out pattern matching of the known IXP prefixes on the traceroute-based data to filter out all IXP routes. It is evident that the parallel CUDA implementation consistently outperforms the serial technique by halving the time required in searching for the prefix hops and returning the route on success. With lesser memory available on the GPU (in comparison to the CPU) we also had to break up our input stream to smaller files which required a higher number of input/output seeks to disk. In spite of these additional operations parallel string matching proves effective in our scenario. To further investigate the working of PFAC and parallel string matching we carried out the same operation but with single prefixes as our search pattern instead of all 373 at the same time. Selecting 20 different prefixes at random we carry out both the serial and parallel searches and we obtain the search times as shown in fig 7.4. Its interesting to observe that serial outperforms the parallel version consistently all the time. Infact the parallel version string matching times for a single prefix are very similar to the time it takes to match all the 373 prefixes. This result gives us valuable insight to the working of the parallel string matching algorithm used in PFAC, which uses a failureless transition scheme. Here with a

single prefix the transition table is much smaller but the steps of splitting and loading the input stream into the GPU memory still takes the same time as before. Moreover another characteristic of IXPs and Internet policy routing affects the results obtained here. Due to the *valley-free* nature of Internet routes, one route could only have one IXP hop in its route. This means with a single pattern to be matched with (and a smaller transition table) every route is matched with a prefix only one at a time while a match with all prefixes (and a larger transition table) would not require repeated memory accesses (per prefix). We thus conclude that the pattern matching in parallel exercise would be effective when searching for all unique patterns simultaneously and not one by one as is done in a general serial implementation with regular expressions (for example with the popular Unix grep tool).

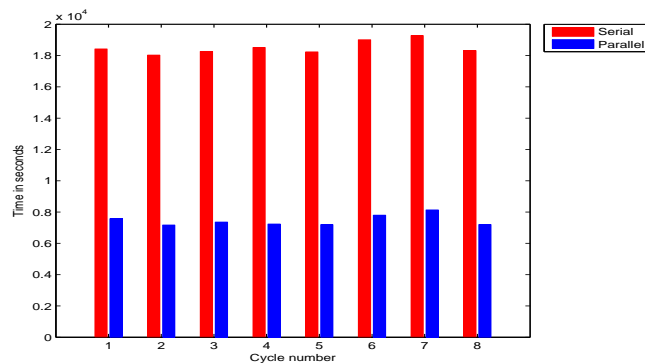


Figure 7.3 Timing comparison between serial and parallel processes for pattern matching in our traceroute data. Times taken to carry out the matching in parallel are consistently less than in serial for every cycle of CAIDA data considered in our experiment (ten).

7.2.4 Diminished PFAC speedup?

From the timing results obtained in the previous subsection we observe the average speedup obtained by using PFAC is about 2x for large traceroute datasets while Lin et al. in [73] show upto 4000x speedups in their test scenarios. The question thus arises as to why our implementation provides these very limited gains in speedup? The answer lies in the fact that the nature of our pattern matching requirements greatly underutilizes the capacity of PFAC resulting in lower improvements in speedup. In the original implementation, PFAC works by assigning each byte of the input stream to a new GPU thread to start the search based on the transition table. With the availability of thousands of GPU threads at any given moment, a huge section of the input stream may be searched simultaneously and it finally returns the individual start positions at the input stream where the pattern match occurs. However in our pattern matching case, the input stream consists of multiple lines of patterned traceroute output containing a series of IP addresses followed by the latency value observed at that hop along the route. If an IP matching the known IXP prefix IP is found we need to record the entire path. This path starts from the source IP (at the beginning of the line) to the final hop in the destination AS (at the end of the line). A line break follows after which the next route is available. Since we need to return the entire path on which the pattern match is successful we change the proposed PFAC scheme to save every line being searched in memory and return it on success. As a result we are able to search only **one** line at a time in our experimental setup resulting in a maximum of 500 search threads on the GPU at a time. With so many available threads remaining idle our observed speedup

is severely compromised. The running time for PFAC in our implementation thus becomes bound by the length of each line being searched while the total running time depends on the number of paths present in the traceroute dataset. As shown in figure 7.2 we load 8k lines¹ at a time into the main memory each of which is then sent into the GPU for the IXP pattern match. More efficient CPU architectures may outperform the PFAC implementation in our case but this step shows that the GPU may be used to carry out a relatively efficient pattern match from traceroute datasets and even a third improvement in run-times does provide a reasonable advancement in the entire measurement and analysis process.

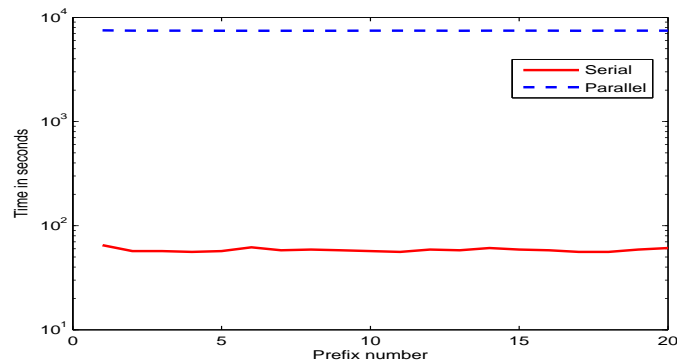


Figure 7.4 Pattern matching per prefix. Here the serial algorithm outperforms the parallel version for every random prefix selected. A single prefix results in a smaller transition table and in this case the net pattern-matching time becomes dependant on the time taken in loading the input data repeatedly to the GPU and executing PFAC.

¹This number is hardware dependent and with more available memory we can increase the number of traceroute logs loaded.

7.3 All pairs shortest path in parallel

Parallel implementations of the Floyd-Warshall APSP algorithm has been a focus of the research community with Harish et al. [39] proposing the use of CUDA in this direction. While the authors implemented a basic version of Floyd-Warshall APSP, subsequent work ([40, 41]) optimized the memory latency and register allocation processes to obtain performance gains. Buluc et al. in [41] implement a recursively partitioned APSP algorithm by casting most operations into matrix multiplications on a semiring. Though there are some added constraints (for example the order of loops cannot be changed arbitrarily) such an approach lends itself to a high degree of speedups, a characteristic exploited by the authors. The authors show a 480x speedup in comparison to a CPU implementation and upto 75x in comparison to an existing GPU implementation. Such a formidable enhancement in performance leads us to use this implementation of APSP on CUDA in our working model to compute the shortest paths in our reference AS delay graph along with a parallel CPU implementation in Matlab DCT. In this section we look at the parallel implementations in detail.

7.3.1 Implementation details

The authors in [41] make available their optimized recursive implementation of APSP which we run on the GPU using CUDA after incorporating some changes. We carry out these changes primarily to ensure that datasets with larger graph sizes may be accommodated in our

measurement framework (8192 vertices was the maximum graph sizes handled in [41]). With our AS graphs extending to 9K or 10K nodes we implement a known clustering algorithm used in the modeling of Internet delay spaces [?] to ensure this requirement for graph sizes is met. The clustering scheme is a *hack* to ensure the APSP implementation of [41] can be used by us. We do not extend their algorithm to accept graphs of larger sizes but instead use existing networking techniques to work around the size limit. We discuss this issue in detail in subsection 7.3.3.

The serial recursive code is the optimized approach as shown by the authors in [41] which provides us with an important baseline to measure and compare our GPU implementation. Matlab provides us with an opportunity for a reference parallel implementation using the DCT which incorporates a whole lot of backend management required in running an algorithm in a distributed fashion.

7.3.2 Timing results

Table 7.1 presents our observations for running times of APSP in our various implementations. The two serial implementations (iterative and recursive) are run on the windows and fedora machines and it can be seen that the windows implementations consistently outperform the linux code. This is primarily due to the fact that the windows machine has a greater clock speed and 2GB of extra RAM memory than the machine running fedora. Secondly, visual studio has advanced memory management features with increased compiler

performance in comparison to the native gcc 4.4.5 running on the Fedora machine. The serial recursive times are close to half that of the iterative version indicating the efficiency of a recursive scheme in the shortest path calculation.

Table 7.1 Timing results (in secs) for the different serial iterative, recursive and different parallel implementations carried out in our experiments.

Num of nodes	Serial Iterative Times		Serial Recursive Times		Parallel Times	
	Windows	Fedora	Windows	Fedora	DCT	GPU
9972	79257	87353	39652	52214	782	31
10250	79812	89103	40002	54287	805	32
10661	81374	91281	44875	58701	811	31
9344	78891	88526	42178	51145	748	28
10441	81365	91892	44992	56488	828	28
10221	81555	89554	46288	56030	886	29
10839	83156	92283	46912	59924	905	28
10058	81617	91221	45764	55585	875	30

The parallel Matlab implementation provides a time improvements which is a couple of orders of magnitude better than the recursive serial implementations. Even though the Matlab DCT is incredibly efficient and the fact that we installed the DCT server on the same

machine as the workers², the parallel implementation shows a couple orders of magnitude in time improvements (in comparison to the recursive serial implementations). What also needs to be noted here is that the Matlab implementation was run on the same machine running the serial versions on Windows, which means the parallel version in itself is efficient.

The recursive GPU times observed are shown in the last column which indicate a significant speedup in comparison to the parallel version. Speedups observed are in the range of 24x-32x is obtained. Speedups are much greater ($\geq 1200x$) when compared to the serial recursive times an even higher if compared with the iterative implementations. However the most relevant comparison here is with the times obtained with the parallel implementation carried out in the Matlab DCT. The recursive GPU implementation provides about 30x speedup while running on the same machine as that running the Matlab DCT denoting a high level of performance increase when we use the GPU.

7.3.3 Graph sizes

As we observe in table 7.1 the number of nodes in our graphs over which we run the APSP algorithm are around 10k. The iPlane dataset from which we obtain our traceroutes monitors Internet paths across these ASes while other measurement systems trace to a varied destination set. With the total number of known ASes in the Internet currently upwards of 30k, the problem of running the APSP scheme for larger graph sizes becomes important.

²The server being on the same machine reduced communication delays with every Matlab worker running an instance of the parallel implementation. This helped us isolate and reduce network communication delays which would increase the total job execution times.

Buluc et al. [41] mention their recursive implementation is valid only for 8192 node graphs, an obvious drawback in our case.

To work around this inherent difficulty, we use properties of Internet topology and path construction to fit our dataset sizes to this known upper limit. Zhang et al. in [?] propose a global clustering metric and algorithm to cluster ASes globally in the Internet delay space by iteratively merging nodes into nearby clusters. The distance between nodes in two clusters is computed by the average delays between the nodes and a cutoff delay value is used to put the bounding condition for stopping the merging process. In our APSP implementation, we carry out this clustering scheme to ensure that the total number of nodes in our graph remains less than the upper limit. By varying the inter-node delay cutoff value we reduce the total number of nodes in our graph by merging multiple nodes into clusters. We realize that this technique does not provide a GPU based solution to the problem of APSP in large graphs, but it does provide us with a relevant and known solution to run Buluc et al's [41] recursive implementation on the GPU with regard to our specific use in determining Internet TIVs. By using the clustering scheme our process can be extended to run on significantly large datasets running into 20-30k nodes, an unlikely scenario since no existing individual Internet measurement infrastructure has such a detailed reach. The drawback in the clustering approach is the loss of granularity due to merging numerous nearby ASes but then again Internet path latencies are in itself estimates dependent on various conditions.

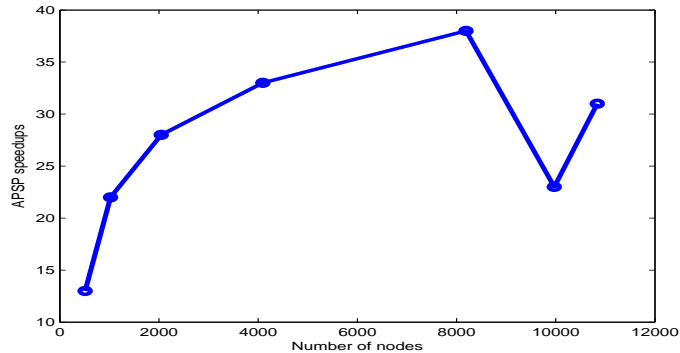


Figure 7.5 Speedups observed in APSP with increasing number of nodes. After the 8192 bound is reached our clustering scheme kicks in which takes a greater amount of time and reduces the overall observed speedup.

We also carry out a relative comparison of the GPU implementation of the APSP computation with the parallel CPU implementation and the timing results are presented in table 7.2. With increasing node sizes a gradual increase in speedups can be observed when compared to or parallel implementation in Matlab (as shown in figure 7.5). The running times for the GPU APSP implementation remains consistent with those reported in [41] but when the graph size increases beyond 8192 vertices our clustering algorithm kicks in to merge close by nodes. To ensure minimal granularity loss we stop the clustering process as soon as the number of vertices in the graph goes below the upper bound, which is why the clustering scheme completes relatively quick (within 10-12s). As shown in the table due to the cluster computation the 9k and 11k node graph APSP speedups is lower than the 8192 node graph, but the obtained speedup still points towards an efficient usage of the GPU. Overall with the clustering technique implemented with respect to our problem, our intuition towards using

the recursive GPU implementation for APSP proves to be an effective step in the overall process of computing Internet delay TIVs.

Table 7.2 Comparing APSP times for graphs of various sizes. Speedups obtained upto 8192 vertices remain consistent with previous work but with sizes greater than 8192, our clustering algorithm first runs to reduce the total number of vertices which in turn almost doubles the total APSP run times and reduces the speedups obtained.

Num of nodes	Parallel DCT	GPU	Speedup
512	.257	.0177	13x
1024	2.101	0.091	22x
2048	8.363	0.299	28x
4096	57.385	1.688	33x
8192	629.31	16.12	38x
9972	781.82	31.35	23x
10839	905.10	28.29	31x

Figure 7.5 also provides an indication of the asymptotic running time of the parallel recursive APSP process. With an increasing number of nodes the speedups observed increase almost linearly upto the upper bound of 8192 vertices after which the clustering scheme takes an increasing amount of time. With graphs much larger than 8k nodes, clustering times start increasing linearly while the graph size is reduced by merging closer nodes together.

Table 7.3 APSP with graph characteristics. Every CAIDA cycle generates equally large graphs with similar average edge weights resulting in very similar APSP completion times.

Cycle Number	Nodes	Edges	Average weights	APSP Times (s)
1257	9972	389805	243.448	31
1261	10250	427144	233.17	32
1269	10661	493121	262.107	31
1270	9344	361480	286.49	28
1271	10441	444507	239.44	28
1272	10221	441017	241.73	29
1273	10839	499583	240.324	28
1274	10058	428457	251.38	28

Thus the speedups obtained can be bound under two different conditions, one under 8k nodes where it increases with increase in graph size (this $O(n)$) while after a tipping point at 8192, the increasing trend again reappears.

7.3.4 Recursive APSP on different AS graphs

As CAIDA's Ark measurement architecture focuses on uncovering greater sections of the Internet from various vantage points across the world, probes are sent to every known prefix and the paths measured. Due to the very nature of this type of Internet measurement

study, every cycle has an approximately similar number of recorded probes. Thus the AS graphs generated from these cycles of data have similar characteristics and are not random in nature. In fact previous Internet topology measurement studies ([26, 74]) have mentioned this AS graph represents the data plane of the Internet due to the underlying mechanism of traceroutes. This means the graphs which we obtain in every cycle exhibit similar underlying characteristics alongwith the obvious ones such as number of nodes, edges and so on. Table 7.3 presents some salient properties of the graphs we obtain and study and it is evident that both the number of nodes and edges in each cycle are consistent. The recursive APSP scheme calculates the shortest paths in similar amounts of time as well. Latencies between every AS is represented as the weight of an edge which too does not vary appreciably or affect the APSP computation times. We present these results to emphasize the fact that the AS graphs derived are not random in nature and are suitable for a recursive parallel shortest path solution.

7.4 Overall results

Combining both the serial and parallel modules, we now look at overall times and gains obtained by implementing the use of GPU's and CUDA in our measurement and analysis framework.

Table 7.5 presents a comparison of the overall time taken in our TIV identification process with both serial and parallel implementations. We select a set of eight cycles of

CAIDA scamper data (some of the cycles missing in the list were not available at the time of our experiment for which we obtained the next available one) all of which are of similar sizes due to the very nature of the Ark measurement infrastructure. As mentioned earlier the entire process is split up into different modules with string search and APSP being identified as candidates for our parallel implementation. In table 7.6 we can observe savings obtained in the first module; the string search to identify IXP routes only within the entire dataset. We observe the parallel implementation decreases the time taken by more than half than that in the serial version. With the 373 IXP prefixes being searched in the traceroute data we are able to separate the routes traversing an IXP with parallel pattern matching quicker than a regular grep based search. Once the IXP routes have been identified the next step of creating the delay graph is a set of serial processes which gives us equal delays values in both implementations before the final shortest path computation. This is where the parallel recursive CUDA implementation makes significant savings on computation time. For graphs with upto 8192 vertices (**CUDA block size * number of threads per block**) the parallel solution generates results in less than a minute while a regular CPU implementation takes more than a day. The CUDA implementation of pattern matching and APSP not only carries out the entire computation with low latency but crucially keeps much of the CPU available for us to carry out other important operations.

Computing speedups obtained for the GPU implementation, we see that the speedup obtained from the serial version is approximately 3-6x while it averages between 1-2x for the parallel implementation. Considering that two large modules in the entire process is serial,

the final speedups are helpful in identifying the TIVs for further analysis. While individually the authors of PFAC show a potential of a 4000x speedup compared to serial approaches (and a 3x speedup from parallel approaches) our pattern matching step reduces total times by a third. This is due to restraints based on our requirements while the APSP module provides us with speedups upto 40x. The serial modules in the whole process reduces the overall speedup but it is nonetheless an improvement considering the length of the entire process. For example if we consider cycle 1257, the best possible parallel CPU implementation would take approximately 5 hours to complete while the GPU implementation finishes in about 3 hours.

Table 7.4 Running times (in secs) for increasing number of nodes and routes. The number of routes measured is increased and the number of visible ASes in the routes denote the nodes. The effect of the serial modules of the TIV identification algorithm control overall times observed.

Nodes	Routes	Parallel	GPU	Speedup
478	10K	246	215	1.14x
1045	50K	474	391	1.21x
1771	100K	733	585	1.25x
5229	1M	5771	4182	1.37x
7926	5M	9837	6807	1.45x
9972	9M	18148	13489	1.34x

With a new cycle of CAIDA traceroutes being generated every 48 hours, the time saved every day in the analysis helps us to carry out passive measurements³ in real time before the new traces are available. With the transient nature of Internet link delays, route latencies may change from hour to hour but a quicker analysis process helps identify short/long lived TIVs and their characteristics.

The nature of TIVs requires the presence of a high number of ASes (nodes in our graphs) in the traceroute to identify greater instances of a violation occurring since more nodes lead to a greater number of paths between nodes. However we analyse the entire running time for one cycle in detail to estimate asymptotic running times for our process for an increasing number of routes.

Table 7.4 presents the running times for an increasing number of routes analyzed. It is very evident that the serial modules in our process control the overall times since these processes in itself are dependent on the number of routes being analyzed. More routes requires greater amounts of paths being parsed, loaded into memory, greater splits to be sent to the GPU for PFAC, greater IP to AS mapping required before finally the APSP is carried out. In spite of this, the speedups shown in 7.4 do show a slight gradual increase with increasing number of routes as the GPU enhancements for modules 2 and 4 start to be more apparent, but their net effects are not extreme. 8192 nodes does prove to be an upper bound as higher number of nodes lead to global clustering increasing the APSP run times.

³Passive measurement systems do not add traffic to the network unlike in active measurements where packets are specifically introduced into the network and their effects measured instantly.

Nevertheless, increasing speedup indicates lower overall runtimes indicating more efficient execution of our TIV identification algorithm.

Table 7.5 Comparing total run times (in secs) for the entire process. Refer to table 7.6 for breakup of individual modules. Speedups for the proposed implementation using GPU's range from 3x-6x versus a serial implementation while in comparison to the best parallel implementation speedups range between 2-4x.

Cycle	Serial	Parallel CPU	GPU
1257	63962	18148	13489
1261	61741	15709	10205
1269	67433	16208	10399
1270	64334	15581	10168
1271	67873	15718	10129
1272	68964	15991	10855
1273	64327	15834	11182
1274	68612	16502	10252

Table 7.6 Table detailing overall performance results. All times in seconds. The route search components takes a third less time while APSP decreases processing times from a day to less than a minute.

Cycle	Module Num	Serial	Parallel	GPU
1257	1	2733	2733	2733
	2	18425	11481	7573
	3	3152	3152	3152
	4	39652	782	31
	Total	63962	18148	13489
1261	1	2786	2786	2786
	2	18739	11904	7173
	3	214	214	214
	4	40002	805	32
	Total	61741	15709	10205
1269	1	2804	2804	2804
	2	19552	12391	7362
	3	202	202	202
	4	44875	811	31
	Total	67433	16208	10399
Continued on next page				

Cycle	Module Num	Serial	Parallel	GPU
1270	1	2718	2718	2718
	2	19250	11927	7234
	3	188	188	188
	4	42178	748	28
Total		64334	15581	10168
1271	1	2721	2721	2721
	2	19979	11988	7199
	3	181	181	181
	4	44992	828	28
Total		67873	15718	10129
1272	1	2850	2850	2850
	2	19648	12077	7798
	3	178	178	178
	4	46288	886	29
Total		68964	15991	10855
1273	1	2841	2841	2841
	2	19397	11911	8136
	3	177	177	177
Continued on next page				

Cycle	Module Num	Serial	Parallel	GPU
	4	46912	905	28
	Total	69327	15834	11182
1274	1	2844	2844	2844
	2	19829	12608	7203
	3	175	175	175
	4	45764	875	30
	Total	68612	16502	10252

CHAPTER 8: IXP ROUTING PERFORMANCE

With exchange points definitely affecting the Internet topology evolution as detailed in the previous chapters, there is a clear need to study the effects of these IXPs on inter-domain routing performance. Are these exchange points actually providing ASes better routes? Design considerations typically point towards lower end-to-end latency for shorter, local routes not requiring inter-continental transit; but does that mean IXP effects are only going to be prominent in local traffic? Are paths through IXPs any different from those not traversing them? In this chapter we look towards answering these questions and determining the effectiveness of end-to-end inter-domain routing through IXPs.

The network benefits of peering through IXPs worldwide has not been studied earlier and this section of our work presents a first step in this direction. While there are numerous metrics characterizing Internet paths (bandwidth, loss rates and throughput to name a few), the defining characteristic is always the end-to-end latency. In this work we first measure latencies between end-hosts along a path where an intermediate hop traverses an IXP (we call such a route an *IXP path* throughout the rest of the paper). These IXP path latencies are then compared to the latencies obtained from an extensive set of alternate paths between the same end-hosts but with the IXP hop being bypassed. Identifying valid alternate paths is an involved process which is first described in detail before we carry out our path comparison

studies. Our measurement framework identifies a default IXP path and its latency, obtains a set of valid alternate paths between the same end-hosts but not traversing the IXP hop, estimates the latencies of these alternate paths and finally compares these to the default latencies.

8.1 Path Selection

To infer the effectiveness of IXP paths [75] requires the identification of a path traversing an IXP, an operation for which traceroute is most suitable. Traceroute not only helps us identify the IXP hop along the route, it also provides an estimate of the end-to-end latency for the path between the measured source-destination AS pair (referred to as *srcAS* and *destAS* respectively). However isolating the IXP hop's effects on this path requires the identification of alternate paths between the same source and destination hosts which are not traversing the IXP network. Estimating these valid alternate paths excluding the IXP is an involved process which we describe (later on) in this section.

8.1.1 IXP path selection

Peering databases such as PCH [1] and PeeringdB [15] along with the IXP Mapping project [16] publish a set of prefixes which are known to belong to major IXPs across the world. By searching for IP addresses within these known prefixes along a traceroute enables identifi-

cation of the IXP paths. Participant ASes at the IXP network are identified by extracting IP addresses of the hops just before and after the IXP IP (as described in [17]) and then mapping the IP address to its corresponding AS number¹. The end to end latency estimate for these default IXP paths are obtained from the completed traceroutes to the destination host and recorded.

8.1.2 Alternate paths: Common provider direct - Type 1

The primary economic benefits of peering for the pair of participating ASes comes from avoiding transit costs to their respective providers for traffic meant for each other. In the case of no peering agreement between the two, traffic would be routed towards the destination through a provider AS (which could be a bigger ISP, transit provider or just a bigger AS with routes to the destination). We exploit this concept towards identifying potential alternate paths towards the destination.

Once the participant ASes (say P_1 and P_2) are identified, providers of these ASes are determined from a combination of CAIDA's AS relationship dataset [76] and BGP tables along with Gao's AS relationship inference algorithms [77]. We then select those ASes which are common to both P_1 and P_2 . Finding the common set of providers ensures the alternate path being identified is not greatly dissimilar to the default IXP path and that a path would surely exist from these providers to the destination AS (via P_2). For large numbers

¹IP to AS mapping is itself an in-exact process. We use the widely used Team-Cymru database [69] to obtain the AS number(s) for a particular IP.

of available common provider ASes, we select twenty (Section 8.4.5 provides the rationale behind selecting this value) random ASes and construct paths via these providers to the destination. The final path is thus an aggregation of the following components:

- Component 1: $srcAS -> P_1$
- Component 2: $P_1 -> T_i$
- Component 3: $T_i -> destAS$

where $T_i = \{T_1, T_2, T_3, \dots, T_k\}$ is the set of common providers of P_1 and P_2 , $1 \leq i \leq k$ and $k = 20$ when number of common providers is greater than 20.

Figure 8.1 depicts the construction of these paths.

8.1.3 Alternate paths: Common provider indirect - Type 2

These paths are designed to isolate the IXP hop from the IXP path and so an additional component between the set of common providers and P_2 is created. The path upto P_1 remains the same as the original, then common providers of participant ASes are determined (as mentioned in the previous subsection) with component 2 comprising of hops between P_1 to the selected common provider. Component 3 is then the path between the common provider to P_2 and the path is rounded off with the exact same hops between P_2 and $destAS$ as in the default IXP path. The final path is thus an aggregation of the following four components:

- Component 1: $srcAS -> P_1$

- Component 2: $P_1 - > T_i$
- Component 3: $T_i - > P_2$
- Component 4: $P_2 - > destAS$

Figure 8.1 presents the extra component and shows how the IXP hop is isolated in the construction of these indirect paths.

8.1.4 Detour Paths

Traditional detour paths [33] have been used extensively in overlay routing to identify the best possible alternate routes to a destination via another source known to the overlay network. In most cases, these paths represent the best available paths between a given set of hosts and as Gummadi et al. show in [57], if a better detour to the default path exists, the detour can be constructed via a single random hop. We use the best available detour path latency for a given IXP path as a benchmark to compare efficiencies of the set of alternate paths. A detour path is shown in figure 8.1 where a default path from another source to the same destination comprises of the second detour component.

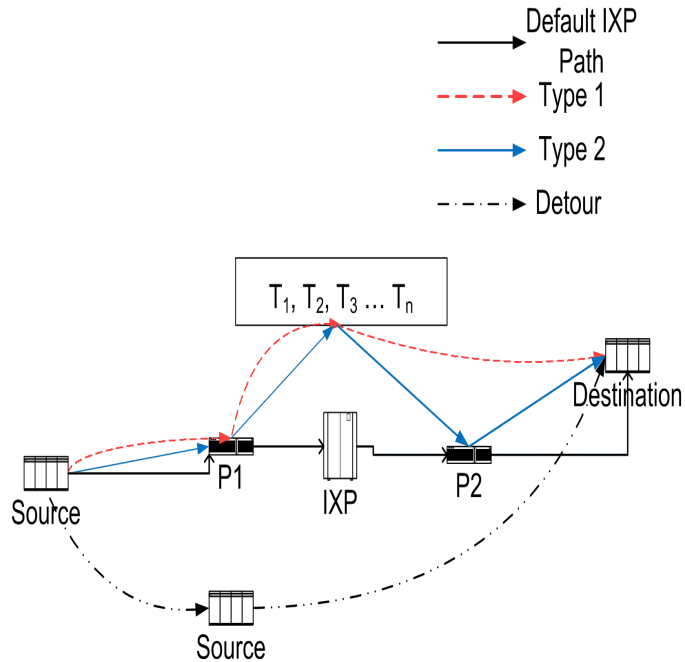


Figure 8.1 The various paths measured in our framework between a set of end hosts. The default path traverses an IXP while the detour comprises of another default path through a different source to the same destination. Type 1 and 2 alternate paths comprise of individual components of the common providers of the IXP participant ASes.

8.1.5 Policy Compliance

The diversity of inter-AS routing policies always makes creation of these *synthetic* alternate paths a process bound to a high degree of uncertainty with respect to their actual validity. Creating these end to end AS paths as an aggregate of smaller components requires us to check if these paths violate an important characteristic such as valley-free routing. For the large number (close to one million) of alternate paths constructed, we run Gao’s algorithm [77] to check for the valley-free property and observe only less than 2% of violations occurring;

a pointer that our technique of aggregating components to create alternates is realistic and valid.

8.2 Measurement Framework

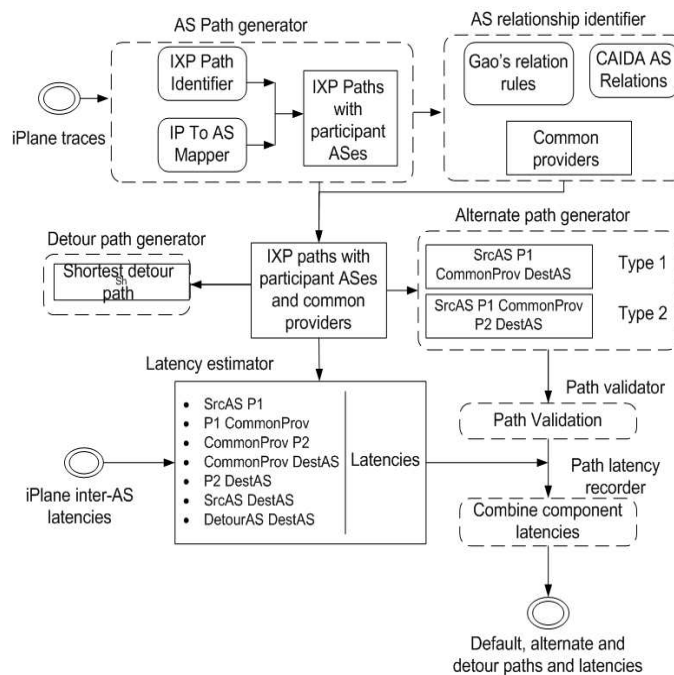


Figure 8.2 System model depicting every component in the proposed framework infrastructure. Traceroutes and inter-AS latencies from iPlane are the starting input files from which the IXP paths are identified with their participant ASes, their common providers obtained, alternate paths generated, validated and finally all the latencies recorded.

Figure 8.2 presents the high-level architecture of our measurement framework. It is a combination of a set of components working with each other to first identify IXP paths from iplane traces and then the different types of alternate paths and their latencies.

8.2.1 Dataset selection

Identifying IXP paths requires per hop information of an Internet route, which is why we look at available traceroute based datasets. CAIDA’s Ark infrastructure [66] and the iPlane measurement project [13] are two popular sources of a wide variety of traceroute based measurements readily available to the research community. While Ark mainly comprises of traceroutes from a set of 54 CAIDA monitors to an IP in every /24 network, iPlane uses more than 150 PlanetLab vantage points to traceroute to a predefined destination list. An additional characteristic of the iPlane data is the publication of an Internet atlas with inter-AS link latencies at the Points of Presence (PoP) level which is extremely relevant to a study such as ours. As shown in [18] a high percentage of plausible inter-AS paths may be constructed from the published atlas, a feature which is an integral requirement in our path design. Hence we decide to use iPlane data made available from iPlane for this work while looking to use the data from CAIDA in future work. Important characteristics of the iPlane dataset can be summed up as follows:

- Traceroutes from large set of PlanetLab vantage points to responding hosts. Diverse geographical locations captured.
- Extensive Internet Atlas with IP to PoP mapping, inter-PoP links and Inter-IP links.
- Inter-PoP latencies published daily. We compute inter-AS latencies from this data.
- Internet atlas enables computation of best available detour route.

- Path components can be created with low percentage of policy violations.
- Historical data updated with high frequency (latencies are updated everyday).

8.2.2 AS path generator

The iPlane measurement infrastructure conducts traceroutes to numerous destination networks from a set of PlanetLab locations. These traceroutes are used to create the Internet atlas comprising of inter-AS latency/loss estimates. This module uses the traceroutes from all the vantage points to search for the IXP paths by matching every IP address on a route to the set of known IXP prefixes released in [16]. IXP participant IPs are identified by separating the IP addresses of the hops just before and after the IXP IP; the participant identification process is discussed in detail in [17]. IPs also need to be mapped to their respective ASes, a problem which in itself has received considerable attention in the research community. We implement an IP to AS mapper in which the IPs addresses visible in iPlane is mapped to their corresponding PoPs before the PoPs themselves are mapped to their ASes. We further verify these ASes by querying the Team Cymru database [69] and drop the IPs which do not map to the same AS in both the above steps.

Output: Discovered IXP paths from a source to destination AS via an IXP with its participant ASes.

8.2.3 AS relationship identifier

This module takes as input the participant ASes in each IXP path and computes their immediate common providers (if available). Using a combination of Gao's AS relationship inference rules [77] and CAIDA AS relationship inference dataset [76], a set of common providers of the participant ASes are determined. For instances when this set of providers is large, we limit it to 20 provider ASes by a random selection.

Output: IXP paths with participant ASes and their common providers

$(srcAS, P_1AS, ComProvAS, P_2, DestAS)$.

8.2.4 Detour path generator

This module searches for the best detour path available in the iPlane atlas between the source-destination ASes for every IXP path. The possibility of numerous available detours means we do not need to identify each of them but only the best one. The NetworkX [78] graph library in python enables quick and efficient searching of the shortest paths through large graphs and records the shortest detour between the source and destination ASes.

Output: Shortest detour path between source AS to destination AS.

8.2.5 Alternate path generator

The *Type 1* and *Type 2* alternate paths are generated in this module before their latencies are estimated. Its essential function is to search the atlas for the links between individual path components created from the path generator and relationship identifier. For example a *Type 1* path which can be represented by $(srcAS, P_1AS, ComProvAS, DestAS)$, is broken down into component links such as $(srcAS, P_1AS)$, $(P_1AS, ComProvAS)$ and $(ComProvAS, DestAS)$. Sub-links between pairs of these ASes are then computed from the Internet atlas and the entire path (of greater granularity) recorded.²

Output: *Type 1* and *Type 2* alternate paths with all intermediate hops.

8.2.6 Path validator

The alternate paths created are sent to this module to ensure they do not violate the valley-free routing property. Since the participant provider ASes are selected based on established relationship rules (in the relationship identifier), most alternate paths remain valid. We drop those which violate this property. This module can be further updated with other metrics to ensure alternate paths of specific types are filtered out.

Output: Alternate paths satisfying the valley-free property of Internet paths.

² $srcAS$ =source AS, $destAS$ =destination AS, $IXPIP$ =IP of IXP, P_1AS = Participant 1 AS, P_2AS = Participant 2 AS, $ComProvAS$ = Common Provider AS, $defLat$ = Default IXP path latency, $detourLat$ = Best detour path latency, $Type1Lat$ = Latency of type 1 alternate, $Type2Lat$ = Latency of type 2 alternate

8.2.7 Latency estimator

The last module of the framework uses iplane latency estimates to compute component latencies for the alternate paths. Each path (*Type 1* and *Type 2*) is earlier broken down into components in terms of ASes. These latencies are then estimated and recorded. Component latencies are then combined and finally stored along with the default and best available detour path latencies.

Output: Final result file with each line identifying an IXP path and the corresponding latencies in this format: (*srcAS, destAS, IXPIP, P₁AS, P₂AS, defLat, detourLat, Type1Lat, Type2Lat*)²

8.2.8 Path latency estimation

The default path latencies for IXP paths are obtained directly from the traceroutes files. Estimating the latencies for the detour and constructed paths require greater attention since they are composed of individual components. Every component is made up of a set of AS links, the latencies of which are extracted from the iPlane inter-AS link data.

iPlane provides link latencies between ASes at the PoP level, which leads to a finer granularity and multiple latency values between the same set of ASes ³. Since we are only able to construct our path segments at the AS level, we consider the median latency value for

³such a situation would typically occur for big ASes with multiple PoPs across continents

every unique AS-pair across all their inter-PoP latencies. We recognize that this approach does not provide an accurate representation of the latency between the considered ASes, but certainly provides a reasonable estimate.

Algorithm 2 Latency prediction for alternate paths

Require: Source AS *componentSrcAS* and destination AS *componentDestAS* numbers to compute latency between

```

1: iplaneFile contains AS link latencies in (PoP1,AS1,PoP2,AS2,latency) format
2: if entry exists for componentSrcAS,componentDestAS in iplaneFile then
3:   return Compute median latency and return value
4: end if
5:
6: Direct link unavailable, break path into intermediate links
7: Using first BGP tables and then a python search algorithm, compute path between componentSrcAS to componentDestAS
   as a sequence of hops. Save result in linksFile
8: Initialize finalLatency = 0
9: for all AS1 AS2 pairs in linksFile do
10:  if AS pair entry exists in iPlaneFile then
11:    Compute median latency medianLat
12:    finalLatency+ = medianLat
13:  else
14:    Mark path as incomplete
15:  end if
16: end for
17: return finalLatency: Final latency of path finalLatency

```

Algorithm 2 presents the details of our path latency computation process. Here when the *componentSrcAS* to *componentDestAS* links are directly available in the latencies file, the latency is returned. However as stated in [18], the iPlane atlas contains a high percentage of Internet paths once broken down into smaller components. To identify the total path we first use a RouteViews BGP table to search for intermediate links if available. If its not

available in the BGP table, we carry out a complete search of the iPlane links to obtain the path between *componentSrcAS* and *componentDestAS* as a sequence of AS hops. This algorithm was implemented using python and the Networkx graph library which is capable of searching through large graphs at high speeds. The path returned is then broken down into individual links and the total latency for the path segment computed. If iPlane is unable to estimate the latency for any link on the path, we mark the path as incomplete and drop it from further use.

Path latency estimation works in conjunction with the path identification process described in section 8.1. AS-pairs chosen in the selection process (whether they are *srcAS* – *destAS* or individual path components) are passed to the latency estimation procedure, which then breaks up the larger components into smaller ones according to the iPlane atlas before computing and returning the total latency between the AS pair.

8.3 Overall evaluation

8.3.1 Dataset analyzed

We analyze the same paths identified and generated in our modeling study (described in 9.4) and measure the latencies obtained. The IXP paths are identified using the set of IXP prefixes released in [16] and the corresponding detour, type 1 and 2 alternates generated.

Path latencies are recorded and further analyzed and the details described in the following subsections below.

8.3.2 Metrics

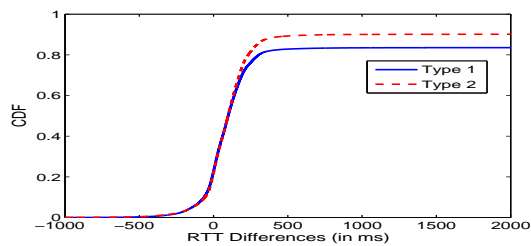
Evaluating the performance of IXP paths with the best available alternates is carried out by using two primary metrics in this study:

RTT (Round Trip Time) Differences: For type 1 and type 2 paths, we compute difference between the default path latency and that of the best alternate. For alternate paths which are better than the default, the difference value is positive. Thus the negative RTT differences denote those paths whose default latencies are the best available. A CDF of the RTT differences enables us to measure the percentage of better/worse paths in our dataset. To compare the computed latencies of the best available alternate with detours, we again compute their RTT difference but with respect to the detour path latency. This is because detours are a well-known artifact of inter-domain Internet routing and generally represent the best paths between a pair of end-hosts. Due to this characteristic, the detour path differences are spread over a greater range of values.

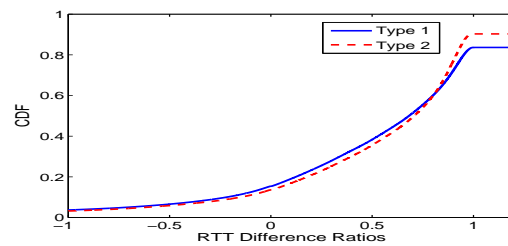
RTT Difference ratios: While the RTT differences provide a direct indication of the spread between the path latencies observed, computing a ratio of the difference computed above with respect to the alternate path latency provides us the magnitude of this difference.

The relative difference ratio is helpful in determining how much better/worse the compared paths are in relation to each other.

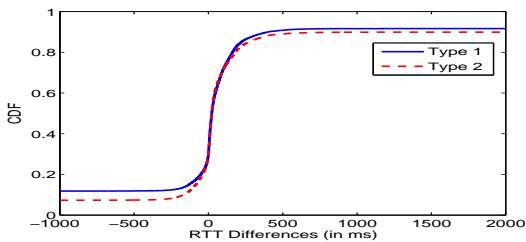
8.3.3 Comparisons with IXP paths



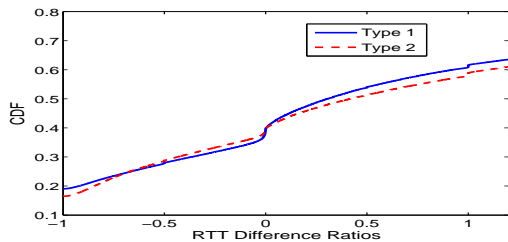
(a) CDF of RTT differences with IXP paths.



(b) CDF of RTT difference ratios with IXP paths.



(c) CDF of RTT differences with detour paths.



(d) CDF of RTT difference ratios with detour paths.

Figure 8.3 CDF of RTT differences and RTT Difference ratios of alternate paths compared to IXP paths. Both types of alternates typically outperform the IXP paths while differences between type 1 and 2 alternates are not very extensive.

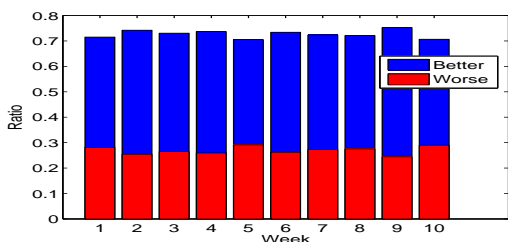
Figures 8.3(a) and 8.3(b) present the CDF of the observed RTT differences between both types of alternate paths and the default path via an IXP. From figure 8.3(a) we may infer that less than 10% of the paths measured exhibit RTT differences less than 0ms which

indicate the default path outperforming the alternate paths. From the remaining path measured the alternates display lower round-trip latencies. Close to 80% of the alternate paths are better by atleast 500ms thereby indicating that the default paths are being slowed down significantly. The type 2 alternates perform marginally better than type 1's but as shown in the difference ratios (fig 8.3(b)), the margin is not by much. Here the percentage of paths at different values of ratios pretty much remain the same indicating the fact that both type 1 and 2 alternates are justifiably better paths instead of the default.

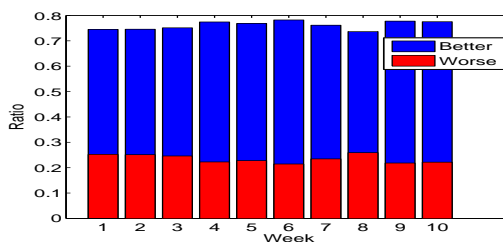
8.3.4 Comparisons with detour paths

Since detour paths have been previously shown to be the best available path from a given source to destination AS, we also compare the alternates with the best available detour in figures 8.3(c) and 8.3(d). Unsurprisingly there are greater number of detours which are significantly better than either alternate path (just less than 20%) with more than 35% of the paths exhibiting RTT difference ratios of close to 0. This is indicative of the fact that alternate paths measured in our framework are not as good as the available detour paths but in fact tend to only isolate the IXP effects, the goal in designing the measurement system. If these alternates were very similar to detour paths in general then comparison with the default paths would not be very effective since detours have been shown to outperform most default Internet paths. Both type 1 and 2 alternates exhibit similar behavior with respect

to the detours and hence serve to reinforce the results of the previous subsection (8.3.3) detailing the comparisons with the default paths.



(a) Type 1 vs Default paths.



(b) Type 2 vs Default paths.

Figure 8.4 Weekly comparison of the percentage of better type 1 and type 2 alternate paths in comparison to the default paths. The ratio of better/worse paths remains consistent over the entire time period of measurement indicating little affects of time on peering performance and routing policies.

8.3.5 Weekly comparisons

The previous subsections show a higher percentage of alternate paths outperforming the default paths for the entire time period of data collection, but it does not provide any indication of a possible skew (for example if any set of IXPs went down for a period of time, or being heavily loaded due to transient traffic effects). To verify overall stability of the measurement process, we compute the percentage of alternate paths exhibiting latencies better/worse than the corresponding default paths for every week. Figures 8.4(a) and 8.4(b) presents these results. For the 10 week period starting from Feb 17,2012 we simply count the

total paths and the number of times each alternate path has a lower latency than the default path. The weekly results are shown in the figures. We observe a constant trend of a higher percentage (approximately 70% for type 1 and higher for type 2) of alternate paths with lower latencies in comparison to the default paths. The percentages do not always add up to 100 since for some paths we are unable to correctly compute the alternate path latencies and are hence dropped. The greater percentage of better type 2 paths reinforce the results discussed in figure 8.3(a) where we infer a greater percentage of type 2 paths displaying RTT differences less than 500ms. Overall, the weekly comparison results show that the number of alternate paths outperforming the default remains pretty much constant over time and is thus not a characteristic of the time period of our measurements.

8.4 Evaluating popular IXPs

We now carry out a smaller study of the nine most popular IXPs based on geographical regions. From publicly available data made available by PCH we select the top 3 IXPs from Europe, USA and Asia-Pacific regions based on the total traffic handled. Using similar metrics mentioned the previous section we measure and analyse IXP path performance in the critical IXPs across the globe.

Table 8.2 The popular IXPs selected and their respective properties (from PCH [1] on 01/22/2012). T=TeraBytes, G=GigaBytes

Region	Name	Prefix range	Traffic	# members	# IXP paths
Asia-Pacific (AP)	Japan Internet Exchange	210.171.224.0/24	251G	85	76187
	Hong Kong Internet Exchange	202.40.160.0/23	193G	105	27269
	Korea Internet Neutral Exchange KINX	192.145.251.0/24	86.2G	42	4428
Europe (EU)	Deutscher Commercial Internet Exchange	80.81.192.0/22 80.81.200.0/24	1.85T	325	431187
	Amsterdam Internet Exchange	195.69.144.0/22 195.69.145.0/24	1.55T	484	422521
	London Internet Exchange	195.66.226.0/23 195.66.224.0/23	1.25T	407	336627

Continued on next page

Region	Name	Prefix range	Traffic	# members	# IXP paths
USA (US)	Equinix IBX Ash- burn	206.223.137.0/24 206.223.115.0/24	305G	72	143431
	New York Interna- tional Internet Ex- change	198.32.160.0/24	191G	137	44022
	Seattle Internet Exchange	206.81.80.0/23	89.6G	151	74368

8.4.1 Dataset details

With iPlane [13] being our chosen dataset, we select an entire cycle of traces conducted from PlanetLab nodes in 162 locations on 01/22/2012 to all destinations in their selected destination list. iPlane uses these traceroutes to construct the Internet atlas for the day and from which inter-AS latencies are estimated. The cycle of iPlane data contains about 20M⁴ routes of which about 3.6M routes traverse one of the known IXPs. Out of these 3.6M IXP

⁴M=Million

paths, 1.55M paths traverse the nine most popular IXPs in the three continents. Of these known IXP paths, about 1.4M IP addresses are successfully mapped to their corresponding ASes via iPlane’s IP-to-PoP mapping, PoP-to-AS mapping files and the team Cymru service [69]. Carrying out this mapping from the IP address to AS is an important step of our process since we estimate inter-AS latencies while computing individual path component latencies.

8.4.2 Selecting popular IXPs

As mentioned in section 8.1.1 various peering databases list details such as prefix allocations, number of participants and total traffic handled for known IXPs worldwide. In this study [79] we look at the three most popular IXPs in Europe, North America and Asia to try to understand the peering effects at these major Internet hubs.

PCH presents a list of worldwide IXPs from which we select the top 3 IXPs per region based on the total traffic exchanged at these locations. Table 8.2 presents the selected IXPs along with the total traffic handled and the number of participants. While an IXP with a greater number of participant ASes could also be considered popular, path latencies are ultimately affected by the total amount of traffic being handled by individual switching networks at the IXPs. Hence we select the top three IXPs per region based on the total traffic transmitted across their networks. As mentioned earlier, out of the 3.6M IXP paths found in the dataset more than 1.55M paths were seen to traverse only the top nine IXPs.

The 43% traffic on these 9 popular IXPs was the driving factor behind our decision to concentrate our attention on only these exchanges in this work.

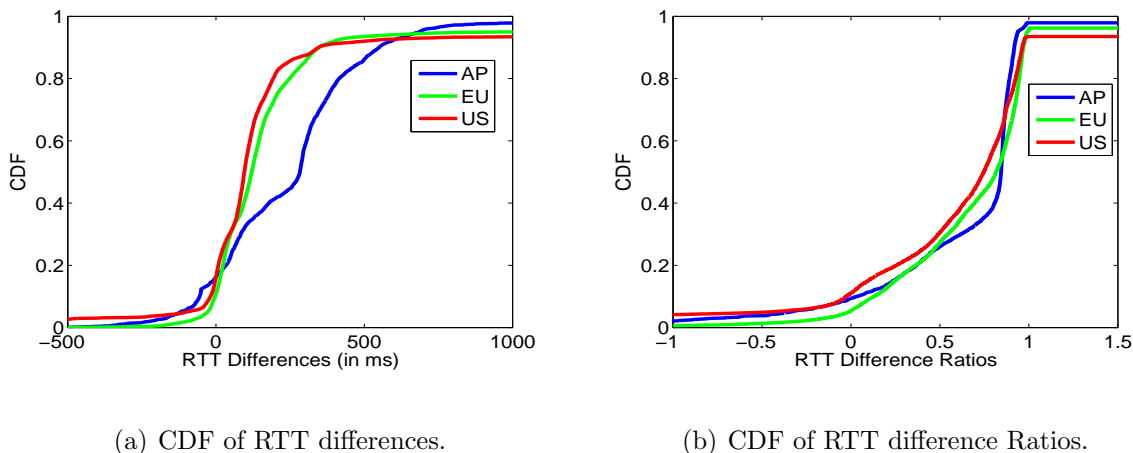
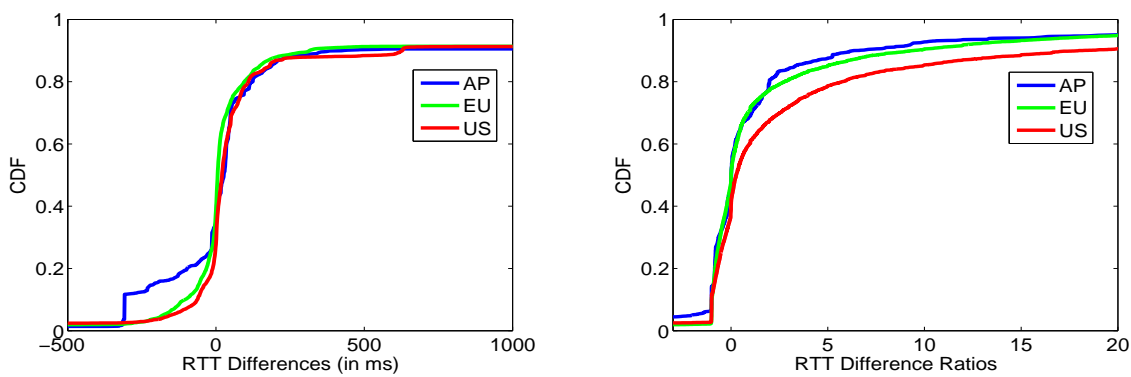


Figure 8.5 CDF of RTT differences and RTT Difference ratios for paths from the common provider to the destination (Type 1). 10% of all paths have a negative difference which means that only one in 10 IXP paths are currently the best available.

8.4.3 Type 1 Paths: Common provider to destination direct

Figures 8.5(a) and 8.5(b) denote RTT differences and difference ratios for the best available type 1 path with respect to the default IXP path latency. 10% of default paths at all locations are better than the available alternates. This is a global characteristic which denotes that one in ten IXP paths are the best available (follow up results show this feature is consistent for all IXP paths). Figure 8.5(a) shows a gradual increase in the latency difference for paths in AP with close to 80% paths exhibiting a higher spread in comparison to the EU and US

paths. This means that the major IXPs in Asia are slowing down traffic significantly more than their counterparts in EU/US; an effect of greater congestion at the exchanges. The difference ratios (fig 8.5(b)) remain greater than half for a high percentage of paths which exhibit the prevalence of quicker paths to the destination from a provider common to both participants.



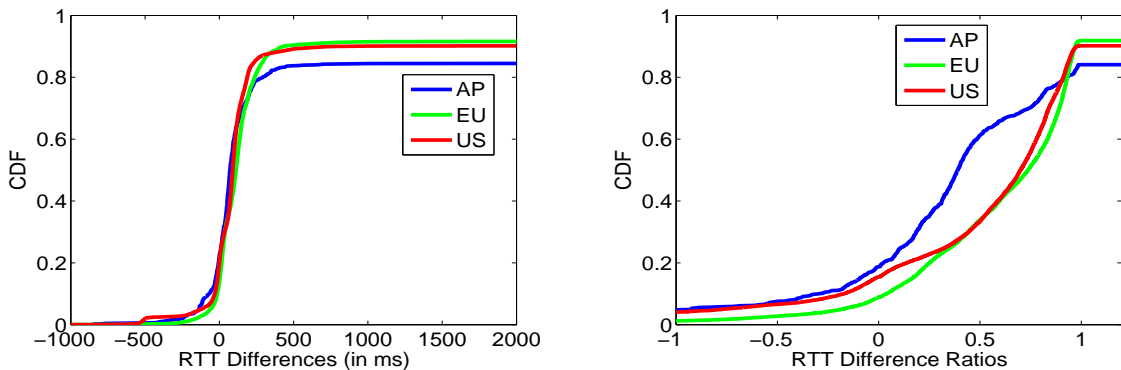
(a) CDF of RTT difference ratios.

(b) CDF of RTT difference ratios.

Figure 8.6 CDF to RTT difference ratios for type 1 paths compared to the best available detour path. 30% of all IXP paths have no better detour but the detours available for the remaining 70% exhibit latency difference ratios considerably larger. This is due to the formation of TIVs in the detour delay space.

Evaluating the metrics for type 1 paths in comparison with the best available detours in figures 8.6(a) and 8.6(b) present some interesting results. Firstly paths across all IXP locations exhibit similar RTT differences/ratios while close to 30% of the paths measured do not have a better alternate detour (paths with RTT differences less than zero in fig 8.6(a)). For the remaining 70% paths, the difference ratios are significantly greater than those in

default IXP paths (see fig 8.5(b)). This signifies that detour paths are significantly better than the computed latencies and is representative of overlay routing in general with the presence of Triangle Inequality Violations (TIVs) [34, 21]. TIVs in the Internet delay space has been attributed to different AS routing policies along with the effects of peering. The creation of these TIVs is uniform as suggested in fig. 8.6(a) with 90% of the paths in AP exhibiting difference ratios of 6 or above. Majority paths through IXPs in the US and EU are lower in comparison but the overall behavior of large TIVs is consistent across all locations measured.



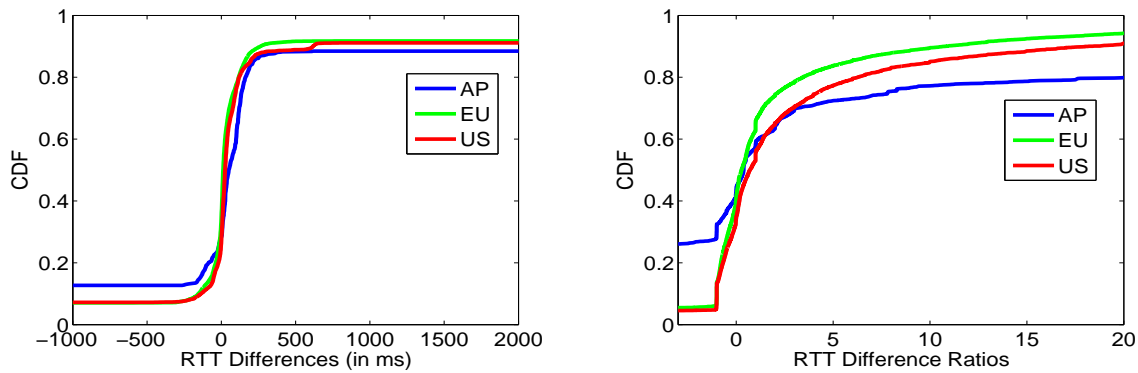
(a) CDF of RTT differences.

(b) CDF of RTT difference ratios.

Figure 8.7 CDF of RTT differences and RTT Difference ratios for paths from the common provider to the destination through the second participant. (Type 2). Paths in AP show greater RTT differences indicating the relative inefficiency of the popular IXPs at these locations. Overall there are greater number of better alternate paths than the default IXP paths.

8.4.4 Type 2 Paths: Common provider indirect

Type 2 paths are those which best isolate IXP effects from the IXP path. As shown in figure 7.7 approximately 10-20% of all the routes exhibit RTT differences less than 0, reinforcing our earlier result that ten percent of the current IXP paths are the most efficient available paths. Figure 8.7(a) shows the RTT differences to be close to or less than zero and the corresponding difference ratios ranging all the way to -1 (fig 8.7(b)). A greater percentage of alternate paths in AP continue to show lower latencies while paths in the US generally outperform those in EU but not by significantly large amounts. The overall picture presents a forceful argument again of the presence of better alternate paths than default IXP paths regardless of location.



(a) CDF of RTT differences.

(b) CDF of RTT difference ratios.

Figure 8.8 CDF of RTT differences and difference ratios for computed latencies from type 2 paths in comparison to the best available detour latencies. Almost 20% of the default IXP paths do not have a better detour available.

RTT differences between the computed latencies for the alternate type 2 paths and the best available detour paths is shown in figure 7. Since the detours are computed from among the iPlane Internet atlas, the set of available detours is limited to the ASes visible in the dataset. Interestingly from figure 8.8(a) we observe approximately 20% of the default IXP paths do not have a better detour available. Such a high percentage (we observe 10% of IXP paths to be the best available in general) can be attributed to the lower sample size of detour path availability in the iPlane atlas. Savage et al. in [33] state the presence of a better detour path for 30-80% of default paths, a characteristic mirrored in this result as well. The difference ratios shown in figure 8.8(b) exhibit the significant savings for those detours with lower latencies with latency difference ratios ranging upto 20 times that of the detour latency (again due to the incidence of large TIVs).

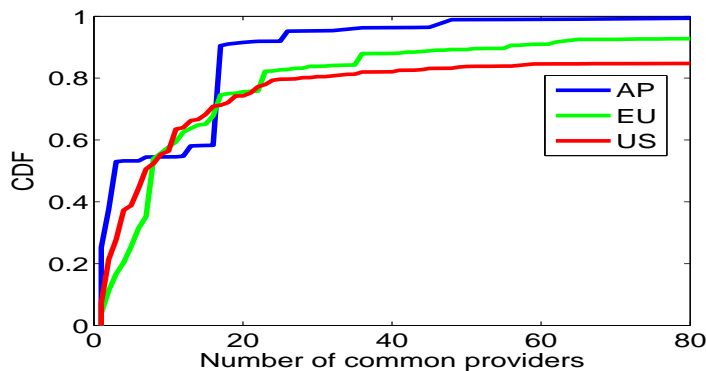


Figure 8.9 CDF of number of common providers for participant ASes. 90% of paths in AP and close to 80% in EU and US have atleast 20 common providers, due to which we select 20 providers randomly from the common provider set for every path in our computation of type 1 and 2 alternate paths.

8.4.5 Common provider characteristics

Building alternate non-IXP paths requires identification of common providers of participant ASes at the IXP hop along the default path. These common providers hold the key to determining if the alternate route is more efficient, which is why we study some characteristics of these ASes in this subsection.

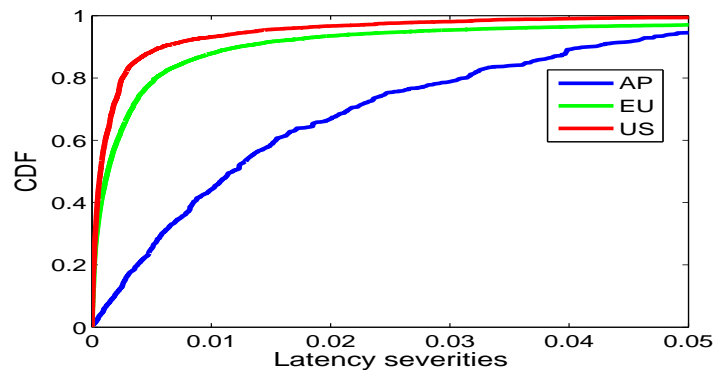


Figure 8.10 CDF of latency severities for type 1 paths. The presence of a large spread between the best path through a common provider in comparison with the other alternate paths in AP results in a higher percentage of paths with greater severity values. Path latencies are more or similar in the other regions.

From participant ASes of every IXP path we derive their common providers and plot the CDF of the number of common providers for every path considered (we drop those paths with no immediate common providers).⁵ Figure 8.9 presents this result from which we observe that this number of providers generally range upto 20 for the AP region and double

⁵Immediate here refers to providers directly one level higher if the tiered hierarchy is imagined. Ultimately the Tier-1 providers are common to all ASes but are only considered if the participant ASes are tier-2.

that for most participant ASes in EU and US. The result helps us determine a threshold of 20 common providers which we then select randomly in our alternate path creation process.

8.4.6 Evaluating provider latencies

With the presence of common providers determined (as shown in the previous subsection) we now evaluate the path latencies through these provider ASes to the destinations. With a maximum of 20 providers being selected for every IXP path we need to define a numeric metric to capture the variance in latencies exhibited through these multiple providers. Wang et al. in [21] propose a TIV severity metric to capture the severity of a particular edge and we use a similar metric in our study. We define the provider latency severities as:

$$Severity = \frac{\sum (Lat_p - Lat_c)/(Lat_c)}{|N|} \quad (8.1)$$

where Lat_c is the computed latency of the best path among all those through the common providers, Lat_p is the current latency for the IXP path and N is the total number of ASes visible in the paths considered. This metric provides a quantitative measure of available provider latencies with higher number of ASes denoting the chances of greater number and length(s) of paths being formed.

Figure 8.10 presents a CDF of latency severities calculated for paths through common providers for type 1 paths. In general a greater percentage of paths in AP exhibit greater severity than the other regions. This is due to the greater spread in available path latencies

through the common providers in AP. Evidently most paths through these providers vary greatly in terms of latencies making optimal path selection a tricky process for the ASes. Paths in the EU and US regions do not vary as much (for more than 90% of the routes, severity values are less than 0.01) and this indicates similar routes to the destination through most providers. It is in AP that routes are more diverse which puts a premium on designing a useful path selection scheme.

It should be mentioned here that we do not report the results of latency severities for type 2 paths in this case as it is essentially very similar to figure 8.10. The severity metric, captures the effects of the different common providers of IXP participants along a IXP path and these common providers remain the same for both type 1 and type 2 paths. The type 2 paths only contain an extra component (the provider to the second participant) on the entire path, which in itself mirrors the effects at the provider. Hence figure 8.10 is representative of the latency severities due to common providers of the IXP participants.

8.5 Limitations

An empirical measurement study such as this is always subject to some limitations which should be considered before definite conclusions can be made. The primary assumption in this study is that latencies between ASes are *estimated* and are not exact. To this end, all Internet measurements are estimates (traceroutes provide estimates of hop latencies, ICMP ping provides estimates of end to end path latency) which point us in the general direction

of what is likely occurring. The latency values used in our computation of individual path component latencies from iPlane are in itself latency estimates carried out in the iPlane measurement infrastructure. Also, our end to end latency estimates are computed between ASes, but it is known that a packet may have to traverse a long distance even after reaching its destination AS (the *Last Mile* problem). ASes also have multiple PoPs which could exhibit vastly different latency values based on the geographic location of the PoP being measured. Our calculations incorporate median latency values for ASes with multiple latencies to get around this problem and try to correctly estimate a representative inter-AS latency. On a related note, IP to AS mapping is an inexact science with broad implications when erroneous. We try to minimize the possibility of errors at this step by using a single dataset to map source, destination and participant IP addresses to their corresponding AS numbers. This is certainly not fool-proof but using the same dataset to carry out the mapping enables consistent lookups of an AS number for an IP prefix.

Table 8.1 Paths analyzed from iPlane date ranges with number of IXP paths found along with the number of alternates generated.(M=Million)

Date	Vantages	IXP paths	Type 1	Type 2	Violations
02-17-12	148	2.63M	89130	88856	6681
02-24-12	151	2.88M	95445	96378	2538
03-03-12	148	2.83M	115962	115531	2217
03-09-12	155	3.08M	129187	122331	5749
03-16-12	153	2.75M	119142	114089	9735
03-23-12	140	2.54M	109613	105309	10932
03-30-12	144	2.49M	120783	115874	6724
04-06-12	144	2.56M	105697	101888	6693
04-13-12	156	2.82M	113357	107807	5784
04-20-12	153	2.96M	122204	117754	4537
04-27-12	140	2.58M	98196	93142	5941

CHAPTER 9: IXP PATH MODELING

The measurement framework defined in the previous section helps identify and determine alternate Internet paths through the IXP participant common providers for the same source-destination AS pair. In this section, we use *Generalized Linear Modeling*(GLMs) [80] to identify a logistic regression model and predict the factors which make an alternate path better than the default IXP path.

9.1 Background: GLM

Standard linear regression models predict the expected value (the *response variable*) as a linear combination of a set of observed values (known as *predictors*) [80]. The response variables in many cases are normally distributed, i.e. they may vary in both positive and negative directions. However for cases where the response variable is *not* normally distributed, such as a binary variable, GLMs are used.

Our measurement of path latency estimates if an alternate path is better/worse than the default IXP path. While the difference between the measured latencies may be modeled, the actual latency difference (in ms) is a transient property with a wide range of possible values. We simplify the problem into a yes/no question: is the type 1 or 2 alternate path

better than the default IXP path? The yes/no choice transforms the response variable to a Bernoulli variable and enables us to identify a model predicting the probability that the alternate path is better than the corresponding IXP path for the given combination of predictors. We explain the rationale and benefits of determining appropriate predictor variables in the following subsection.

9.2 Predictors for the GLM

Modeling the probability that an alternate path is better than the default first requires the identification of the predictors¹. These are essentially characteristics of the alternate paths which affect the overall path latency. Recall that the types 1 and 2 paths use the set of common providers of the participant ASes to construct the best possible alternate. Based on this feature, we identify the following three predictor variables:

Number of (known) IXP participants: Can be summarized as a measure of IXP popularity. Bigger IXPs will have a greater number of participant ASes which in turn would lead to greater traffic exchanged on the IXP network. Higher traffic volume would lead to greater latencies along the IXP hop. Using a variety of sources such as PCH [1], PeeringDB [15] and the individual websites of the IXPs, we compile the number of participant ASes at each IXP monitored.

¹In GLMs there is no specific method of identifying the predictors. An expert with sufficient domain knowledge identifies probable predictors and the model fitting shows if the predictors are useful or not.

Average degree of common providers: The set of immediate common providers of participant ASes are generally bigger ASes who are either solely transit providers or provide routes to other destinations. We measure the size of these providers (in terms of degree) and compute the average degree of the set. The rationale behind this selection is that bigger providers will have a greater number of routes available thereby enabling more choice in creating a better alternate path. Our methodology in computing this common provider average degree is explained in subsection 9.3.

IXP hop location: The IXP hop is typically located closer to the core of the Internet since there is a higher probability of middle sized ASes setting up a peering relationship than a smaller or a stub AS. However if the IXP hop is located closer to the destination, then the alternate path is quite similar to the default path (the overlap of hops from source to participant 1 is quite high) and vice versa for IXP hops closer to the source. We thus divide the IXP path into three sections and denote the IXP hop location as being located within the three sections of the path: first, middle or end of the path.

9.3 Computing provider AS degree

Computing AS size in terms of node degree is a routine task in Internet topology studies. Here large graphs of the Internet are constructed using a variety of publicly available datasets where each node in the graph is an AS and edges between nodes denoting relationships

between the ASes (customer, provider, peering and so on). We create a map of the Internet using the following sources:

- **RouteViews** [7]: Snapshots of BGP routing tables data from the RouteViews project.
- **CAIDA's Ark** [12]: The Archipelago topology infrastructure from CAIDA where a set of monitors across the globe probe every /24 prefix across the Internet.
- **Dimes** [14]: Another popular Internet mapping project which uses traceroute measurements from millions of end-users worldwide to discover and identify a large number of links.
- **iPlane** [18]: Traceroute based probes using PlanetLab to create an Internet atlas of AS links.

Combining the unique AS links (for 30 successive days) obtained from the above data sources, we create an Internet map with the known AS relationships from which we identify the set of common providers. Note that the Internet map we create here need not be totally complete (a common problem in Internet topology studies known as the *missing links* problem) as our primary intention is to identify enough providers common to the participating ASes.

9.4 Generating path data

Using our measurement framework (as described earlier), we generate path comparisons for iplane traceroutes conducted every Friday from the second week of February (2/17/2012) to the end of May (5/25/2012). Thus a total of 15 cycles of paths are analyzed and their respective alternates generated. Every iPlane cycle typically contains traceroutes from approximately 140 PlanetLab vantage points to a wide variety of destinations. We randomly select 3000 default IXP paths (where the source-destination AS pair is unique) from every vantage point, thereby analyzing close to 420K paths every week. Type 1, type 2 and detour paths are generated for every default path analyzed and the predictor variables for each path is recorded. For use in the modeling step, the data is then formatted into different buckets for every range of values and a count of the number of successes recorded. A success is an instance when the alternate path exhibits lower latency than the IXP path. The total number of paths visible within the range of predictor values is also recorded. This is because our model aims to predict the likelihood that an alternate path with the selected predictor variable combination will be better than the default IXP path.

9.5 Identifying best fit

We carry out the generalized linear modeling approach on our data using Matlab's statistical toolbox using the *GeneralizedLinearModel* class. The functions cycle through all possible

combinations of predictor variable values and provide the best possible fit. To simplify the model selection, we only obtain the best possible linear fit for our data². The three predictor variables: number of IXP participants (NumPart), average degree of common providers (AvDeg) and hop location (HopLoc) are divided into buckets and the probability of success fitted. We use the average *R-Square* value for the fitted data as the metric for identifying the best model. It can be defined as the square of the correlation between the response values and the predicted response values exhibiting a range between 0 and 1. R-Squared values closer to 1 indicate a greater proportion of the variance of the residuals³ being accounted for by the predicted model⁴.

Learning the model also requires a cross-validation technique [81] to ensure the analysis results generalize to independent data sets. To ensure that the predictive model performs accurately in practice, we carry out a 5-fold cross-validation process to learn the model. Here the data is randomly partitioned into 5 sub-partitions and the model is trained from a combination of the 4 sub-partitions. The final sub-partition is then used to test and validate the selected model. For every model, the R-Squared value is computed and then tabulated across every partition and finally the model with the highest average R-Squared value is selected to be the best fit. Table 9.1 displays an abbreviated version of the different values obtained from the learning process.

²Quadratic and higher order fits are generally even better but come at a higher computing cost and added complexity.

³Residuals denote the difference between the observed data and predicted values.

⁴For example, an R-Squared value of 0.7281 denotes that the fit explains 72.81% of the total variation in the data.

Table 9.1 R-Squared values for all models with 5 fold cross-validation, x_1 = Number of IXP Participants, x_2 = Average common provider degree, x_3 = IXP hop location; $X_1 : x_2$ denotes an interaction between the respective variables.

Model	P1	P2	P3	P4	P5	Average
$1 + x_1$	0.5523	0.5141	0.5115	0.5271	0.5221	0.52542
$1 + x_1 + x_1 : x_2$	0.7552	0.7448	0.7157	0.7324	0.7588	0.737925
$1 + x_1 + x_1 : x_2 + x_3 : x_1$	0.9359	0.9425	0.9394	0.9274	0.9371	0.93646
$1 + x_1 + x_1 : x_2 + x_3 : x_2$	0.9495	0.9524	0.9491	0.9422	0.9482	0.94828
...
...
$1 + x_1 + x_3 + x_1 : x_2 + x_3 : x_2 + x_3 : x_1$	0.95	0.9527	0.9493	0.9425	0.9487	0.94864
$1 + x_1 + x_3 : x_2$	0.9437	0.9467	0.9433	0.9339	0.9417	0.94186
...
...
$1 + x_2 + x_1 + x_3 + x_1 : x_2 + x_3 : x_2 + x_3 : x_1$	0.9502	0.9527	0.9492	0.9425	0.9487	0.94866
...

From the table we observe the following linear model to exhibit the highest R-Squared average:

$$y = 1 + x_2 + x_1 + x_3 + x_1 : x_2 + x_3 : x_2 + x_3 : x_1 \quad (9.1)$$

where x_1 = Number of IXP Participants, x_2 = Average common provider degree, x_3 = IXP hop location; $x_1 : x_2$ denotes an interaction between x_1 and x_2 .

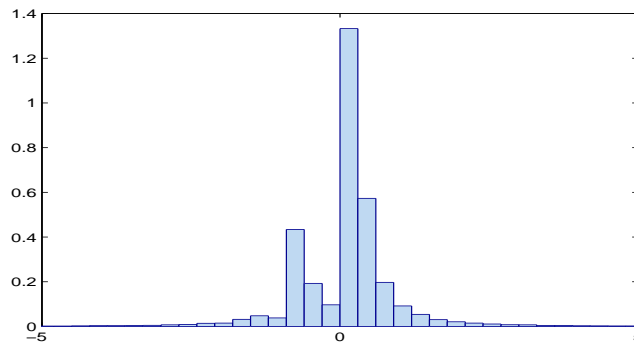


Figure 9.1 Residual plots for the best fit model identified. The histogram shows few values on either side of 0 denoting lower error residuals.

The average R-Squared value of 0.94866 denotes a high fit percentage of 94.86% with the binomial logistic link function. With the model fit being identified, it is now important to verify and identify potential problems in the fit.

We first analyze the residuals from the respective histogram and CDF plots as shown in figures 9.1 and 9.2.

The histogram indicates a high percentage of residuals with very low values centered around zero indicating a good model fit. Analyzing the residuals in more detail by constructing their CDF, we observe distinct tails indicating residuals across the median value.

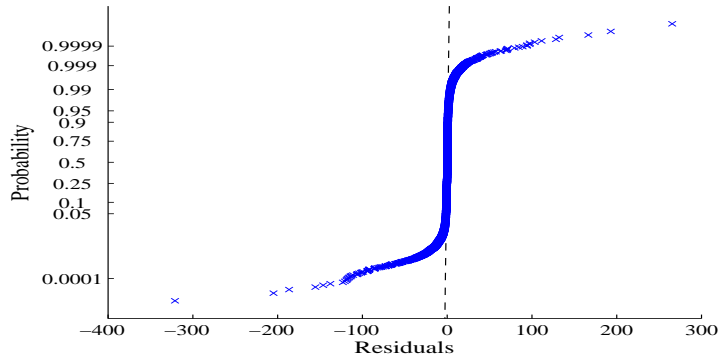


Figure 9.2 The CDF plot indicates the probability of a few outliers being extremely low with most residuals remaining close to zero.

However the probability of the really large residuals are extremely low (with probability lower than 0.05). These values are likely outliers in the data. The CDF reinforces the conclusion that a high percentage of residuals are zero or really close, indicating a good fit.

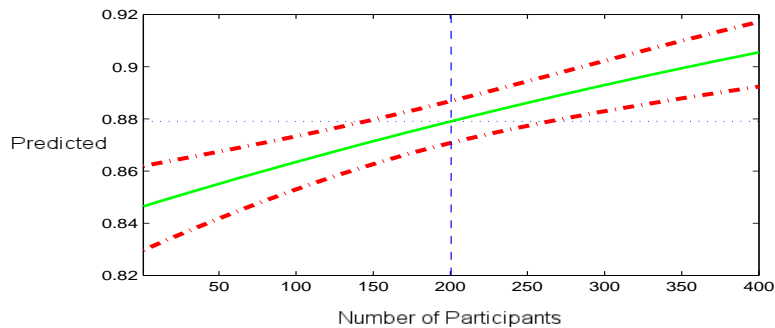


Figure 9.3 Residual plots for the best fit model identified. The histogram shows few values on either side of 0 denoting lower error residuals. - 1

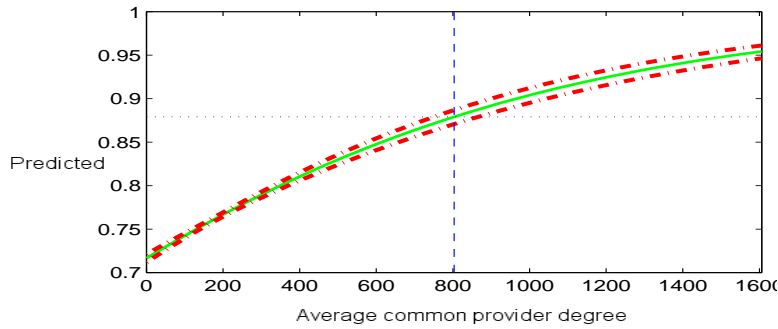


Figure 9.4 The CDF plot indicates the probability of a few outliers being extremely low with most residuals remaining close to zero. - 2

9.6 Predictor variable effects

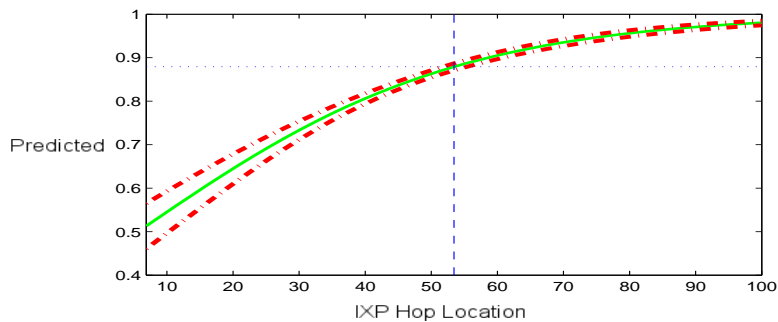


Figure 9.5 The CDF plot indicates the probability of a few outliers being extremely low with most residuals remaining close to zero. - 3

With the best model fit (as described in the previous subsection) identified we now look at individual effects of the predictor variables on the model properties. The series of plots (figures 9.3, 9.4, 9.5) display the predicted value as a function of the single predictor variable within the prediction bounds for the fitted curve. We discuss each of them in detail:

Number of IXP participants: Figure 9.3 displays the predictor value increases with an increase in the number of participants. This behavior is self-evident since an increasing number of participants leads to greater peering opportunities at the exchange points since most ASes tend to peer with *open* peering policies. This leads to greater traffic flowing through the IXP network which in turn increases the probability of greater packet delivery latencies. Thus the probability that an alternate path is better than the IXP path increases. The confidence intervals for the predicted values are however fairly wide indicating the interval values need more data points to obtain a closer fit. However, the problem with participant data from IXPs is that not all the IXPs we monitored had updated participant information readily available. Peering databases such as PCH and PeeringDB do not update participant lists frequently while not all official IXP websites are maintained with the information. The decreased quality of the participant numbers is thus reflected in our model fit with the wider confidence intervals. However, the trend of increasing probability of a better path with increased number of participants is verified in this figure.

Common provider average degree: Larger provider ASes will have a greater number of paths to the destination or faster links to the ASes downstream to the IXP. While average degree is not a very accurate metric to determine the size of an AS, the average degree provides us with a fairly good assessment of its reach. Figure 9.4 confirms the intuition that the larger participant provider ASes increase the probability of finding a better path to the destination. The tight confidence bounds indicate a good fit of the identified model.

IXP hop location: The intuition behind selecting this predictor variable is to determine the effects of the length of a path following the IXP hop to the destination AS. A greater number of hops to be traversed after the IXP would mean the potential for identifying better alternate paths is greater. However figure 9.5 exhibits a different underlying behavior; as the IXP hop moves closer to the destination the probability of obtaining a better path increases. With valley-free Internet paths, IXPs located in the Internet's core and the final two-third's of a path are potentially more loaded due to their increased popularity thereby increasing the probability of a more efficient alternate being present.

9.7 Discussion

Our goal behind identifying a model to predict probability of better alternate paths is twofold: firstly we want to study instances leading to the IXP paths being outperformed by valid alternates and secondly, we aim to identify the underlying factors driving the design of these alternate paths. Based on the results of the previous subsection we observe how the predictor variables are affecting the creation of the probability model. We show that the relation between the observed latencies of IXP paths and the alternates depend significantly on these variables; pointers which should be considered by ASes looking to set up peering relationships. With a definite focus on IXP participant number, size of common providers and IXP hop location on the path, an informed decision as to the network benefit of peering

at the particular IXP may be taken. Identifying these factors which influence the overall efficiency of an IXP path is the primary lesson learnt from our modeling study.

CHAPTER 10: CONCLUSIONS

The Internet AS ecosystem is a constantly evolving and dynamic system driven by the economics of the various players in the market. Transit providers, small and large ISPs, content providers, businesses and home users are all vital components of this network ecosystem in which underlying economics are the driving forces behind its evolution. For example a larger ISP provides routing services to smaller customer ISPs in exchange for payments. The customers need the provider to service their own customers. Now when two or more customers observe that they can save significant costs by bypassing their provider, they would typically invest in setting up peering hardware and exchange traffic along the newly created peering link. The birth, death and rewiring of inter-AS logical links are thus motivated by the underlying economics of business relationships between various ASes. The primary goal of commercial organizations and corporations is to maximize profits and hence the revenues.

In this work, we design and implement a measurement framework to infer path latencies of alternate paths isolating the IXP effects. We consistently observe the high rate of over-utilization of default Internet paths through the public exchange points. We observe that one of ten IXP paths is the best available path amongst all other Internet paths, a characteristic indicating the potential of proper planning in the design and selection of an IXP for a peering relationship between participating ASes. Providing savings in end to end

path latencies, the technical goal of an exchange point, may be improved significantly. The data generated from the proposed framework is also used to model the underlying dynamics of paths through the IXPs and help predict the availability of better alternate paths. The goal here was to identify these factors which influence the probability that an available alternate path exists which may outperform the IXP path. We observe that the number of IXP participants, size of the common providers of these participants and the relative location of the IXP along the path exhibit a direct influence on the alternate path's performance. Modeling these path effects is helpful for ASes to decide on an appropriate peering location and help in making a qualitative decision of the latency benefits available from carrying out a peering decision.

We also study the most popular IXPs in three worldwide regions which are responsible for the efficient transfer of a huge amount of peering traffic. While participating ASes at these IXPs are generally networks/ISPs of a moderate size and closer to the core, they have a number of common providers which possess a better path to the destination AS. We find that the percentage of better paths is greater in Asia in comparison to European and US exchanges. These alternate paths are also numerically more efficient in terms of latencies than their counterparts in the other regions which is mainly due to the lower amounts of traffic being handled daily at these locations.

Overall, this dissertation presents useful insight into the workings of route dynamics at the exchange points across the world. The switching networks at these locations are responsible for huge amounts of traffic everyday and play a major role in determining network

services for millions of end-users. By pointing out the potential for improvements at these major locations, the lessons learnt here will be applicable to a large cross-section of ASes comprising of the peering fabric of the Internet.

LIST OF REFERENCES

- [1] “Packet clearing house.” <http://www.pch.net>.
- [2] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, “The flattening internet topology: natural evolution, unsightly barnacles or contrived collapse?,” in *PAM’08: Proceedings of the 9th international conference on Passive and active network measurement*, (Berlin, Heidelberg), pp. 1–10, Springer-Verlag, 2008.
- [3] A. Dhamdhere and C. Dovrolis, “The internet is flat: modeling the transition from a transit hierarchy to a peering mesh,” in *Proceedings of the 6th International COncference, Co-NEXT ’10*, (New York, NY, USA), pp. 21:1–21:12, ACM, 2010.
- [4] M. Z. Ahmad and R. Guha, “Understanding the impact of internet exchange points on internet topology and routing performance,” in *Proceedings of the ACM CoNEXT Student Workshop, CoNEXT ’10 Student Workshop*, (New York, NY, USA), pp. 18:1–18:2, ACM, 2010.
- [5] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, “Anatomy of a large european ixp,” in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, SIGCOMM ’12*, (New York, NY, USA), pp. 163–174, ACM, 2012.
- [6] Y. He, G. Siganos, M. Faloutsos, and S. Krishnamurthy, “Lord of the links: a framework for discovering missing links in the internet topology,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 391–404, 2009.
- [7] “Routeviews routing table archive.” <http://www.routeviews.org>.
- [8] B. Huffaker, Y. Hyun, D. Andersen, and K. Claffy, “Skitter as links dataset - jan 2000 to may 2009.” http://www.caida.org/data/active/skitter_aslinks_dataset.xml.
- [9] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, “In search of the elusive ground truth: the internet’s as-level connectivity structure,” *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 1, pp. 217–228, 2008.
- [10] R. V. Oliveira, B. Zhang, and L. Zhang, “Observing the evolution of internet as topology,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 313–324, 2007.

- [11] B. Zhang, R. Liu, D. Massey, and L. Zhang, “Collecting the internet as-level topology,” *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 1, pp. 53–61, 2005.
- [12] E. A. Young Hyun, Bradley Huffaker and M. Luckie, “The caida ipv4 routed /24 topology dataset - jan 2009.” http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml.
- [13] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, “iplane: an information plane for distributed services,” in *OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation*, (Berkeley, CA, USA), pp. 367–380, USENIX Association, 2006.
- [14] Y. Shavitt and E. Shir, “Dimes: let the internet measure itself,” *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 71–74, 2005.
- [15] “Public exchange points search/list.” http://www.peeringdb.com/private/exchange_list.php.
- [16] B. Augustin, B. Krishnamurthy, and W. Willinger, “IXPs: mapped?,” in *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, (New York, NY, USA), pp. 336–349, ACM, 2009.
- [17] K. Xu, Z. Duan, Z.-L. Zhang, and J. Chandrashekar, “On properties of internet exchange points and their impact on as topology and relationship,” *Networking*, vol. 3042, 2002.
- [18] H. V. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, and A. Venkataramani, “A structural approach to latency prediction,” in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, (New York, NY, USA), pp. 99–104, ACM, 2006.
- [19] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, “A knowledge plane for the internet,” in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, (New York, NY, USA), pp. 3–10, ACM, 2003.
- [20] H. Zheng, E. K. Lua, M. Pias, and T. G. Griffin, “Internet routing policies and round-trip-times,” in *PAM*, pp. 236–250, 2005.
- [21] G. Wang, B. Zhang, and T. S. E. Ng, “Towards network triangle inequality violation aware distributed systems,” in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, (New York, NY, USA), pp. 175–188, ACM, 2007.
- [22] C. Lumezanu, D. Levin, and N. Spring, “Peerwise discovery and negotiation of faster paths,” in *ACM Sigcomm Workshop on Hot Topics in Networking*, 2007.

- [23] C. Ly, C.-H. Hsu, and M. Hefeeda, “Improving online gaming quality using detour paths,” in *ACM Multimedia*, pp. 55–64, 2010.
- [24] R. Govindan and A. Reddy, “An analysis of internet inter-domain topology and route stability,” in *INFOCOM ’97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 2, pp. 850–857 vol.2, Apr. 1997.
- [25] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat, “Systematic topology analysis and generation using degree correlations,” in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM ’06*, pp. 135–146, 2006.
- [26] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K. Claffy, and A. Vahdat, “Lessons from three views of the internet topology,” Aug 2005.
- [27] H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger, “Towards capturing representative as-level internet topologies,” *Computer Networks*, vol. 44, no. 6, pp. 737–755, 2004.
- [28] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, “The (in)completeness of the observed internet as-level structure,” *Networking, IEEE/ACM Transactions on*, vol. 18, no. 1, pp. 109–122, 2010.
- [29] Y. He, G. Siganos, M. Faloutsos, and S. V. Krishnamurthy, “A systematic framework for unearthing the missing links: Measurements and impact,” in *NSDI*, 2007.
- [30] T. Bu and D. Towsley, “On distinguishing between internet power law topology generators,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002.
- [31] S. Zhou and R. J. Mondragón, “Accurately modeling the internet topology,” *Phys. Rev. E*, vol. 70, p. 066108, Dec 2004.
- [32] E. Gregori, A. Improta, L. Lenzini, and C. Orsini, “The impact of ixps on the as-level topology structure of the internet,” *Computer Communications*, vol. 34, no. 1, pp. 68–82, 2011.
- [33] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, “The end-to-end effects of internet path selection,” *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 289–299, 1999.
- [34] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee, “Triangle inequality and routing policy violations in the internet,” in *Proceedings of the 10th International Conference on Passive and Active Network Measurement, PAM ’09*, (Berlin, Heidelberg), pp. 45–54, Springer-Verlag, 2009.

- [35] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, “Vivaldi: a decentralized network coordinate system,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 15–26, August 2004.
- [36] N.-F. Huang, H.-W. Hung, S.-H. Lai, Y.-M. Chu, and W.-Y. Tsai, “A gpu-based multiple-pattern matching algorithm for network intrusion detection systems,” in *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops*, (Washington, DC, USA), pp. 62–67, IEEE Computer Society, 2008.
- [37] R. Smith, N. Goyal, J. Ormont, K. Sankaralingam, and C. Estan, “Evaluating gpus for network packet signature matching,” in *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pp. 175–184, april 2009.
- [38] A. V. Aho and M. J. Corasick, “Efficient string matching: an aid to bibliographic search,” *Commun. ACM*, vol. 18, pp. 333–340, June 1975.
- [39] P. Harish and P. J. Narayanan, “Accelerating large graph algorithms on the gpu using cuda,” in *HiPC*, pp. 197–208, 2007.
- [40] G. J. Katz and J. T. Kider, Jr, “All-pairs shortest-paths for large graphs on the gpu,” in *Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, GH ’08, (Aire-la-Ville, Switzerland, Switzerland), pp. 47–55, Eurographics Association, 2008.
- [41] A. Buluç, J. R. Gilbert, and C. Budak, “Solving path problems on the gpu,” *Parallel Comput.*, vol. 36, pp. 241–253, June 2010.
- [42] M. Z. Ahmad and R. Guha, “Studying the effects of internet exchange points on internet topology,” *Journal of Information Technology and Software Engineering (Accepted)*, 2012.
- [43] M. Ahmad and R. Guha, “Impact of internet exchange points on internet topology evolution,” in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pp. 332–335, oct. 2010.
- [44] Y. Hyun, A. Broido, and K. C. Claffy, “Traceroute and bgp as path incongruities,”
- [45] 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*, (New York, NY, USA), pp. 251–262, ACM, 1999.
- [46] H. Haddadi, S. Uhlig, A. Moore, R. Mortier, and M. Rio, “Modeling internet topology dynamics,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 65–68, 2008.

- [47] A.-L. Barabasi and R. Albert, “Emergence of Scaling in Random Networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [48] A. Lakhina, J. Byers, M. Crovella, and P. Xie, “Sampling biases in ip topology measurements,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003.
- [49] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore, “On the bias of traceroute sampling: or, power-law degree distributions in regular graphs,” in *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 694–703, ACM, 2005.
- [50] W. Willinger, D. Alderson, and J. C. Doyle, “Mathematics and the Internet: A Source of Enormous Confusion and Great Potential,” *Notices of the AMS*, vol. 56, pp. 586–599, May 2009.
- [51] M. E. J. Newman, “Assortative mixing in networks,” *Phys. Rev. Lett.*, vol. 89, p. 208701, Oct 2002.
- [52] S. Zhou and R. Mondragon, “The rich-club phenomenon in the internet topology,” *Communications Letters, IEEE*, vol. 8, pp. 180 – 182, mar. 2004.
- [53] M. Gaertler and M. Patrignani, “Dynamic analysis of the autonomous system graph,” in *in IPS 2004, International Workshop on Inter-domain Performance and Simulation*, pp. 13–24, 2004.
- [54] “A faster algorithm for betweenness centrality,” vol. 25, no. 2, pp. 163–177, 2001.
- [55] H. Haddadi, D. Fay, S. Uhlig, A. W. Moore, R. Mortier, and A. Jamakovic, “Mixing biases: Structural changes in the as topology evolution,” in *TMA*, pp. 32–45, 2010.
- [56] D. Fay, H. Haddadi, A. Thomason, A. W. Moore, R. Mortier, A. Jamakovic, S. Uhlig, and M. Rio, “Weighted spectral distribution for internet topology analysis: theory and applications,” *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 164–176, 2010.
- [57] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall, “Improving the reliability of internet paths with one-hop source routing,” in *OSDI'04: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation*, (Berkeley, CA, USA), pp. 13–13, USENIX Association, 2004.
- [58] M. Z. Ahmad and R. Guha, “A measurement study determining the effect of internet exchange points on popular webservers,” in *Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks, LCN '10*, (Washington, DC, USA), pp. 976–982, IEEE Computer Society, 2010.
- [59] M. Z. Ahmad and R. Guha, “Effects of popular online destinations through public exchange points on internet route latencies,” *Journal of Networks (Accepted)*, 2012.

- [60] “Planet lab website.” <http://www.planet-lab.org>.
- [61] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang, “Locating internet bottlenecks: algorithms, measurements, and implications,” in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 41–54, ACM, 2004.
- [62] N. Hu, L. (erran Li, Z. M. Mao, P. Steenkiste, and J. Wang, “A measurement study of internet bottlenecks,” in *In Proc. IEEE INFOCOM*, pp. 1689–1700, IEEE Press, 2005.
- [63] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, “User-level internet path diagnosis,” in *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, (New York, NY, USA), pp. 106–119, ACM, 2003.
- [64] V. Paxson, “End-to-end internet packet dynamics,” *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 4, pp. 139–152, 1997.
- [65] Y. Zhang, V. Paxson, and S. Shenker, “The stationarity of internet path properties: Routing, loss, and throughput,” tech. rep., In ACIRI Technical Report, 2000.
- [66] “Caida: Archipelago measurement infrastructure.” <http://www.caida.org/projects/ark/>.
- [67] M. Ahmad and R. Guha, “Analysing global triangle inequality violations due to internet exchange points for future overlay networks,” in *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, pp. 1082–1089, oct. 2012.
- [68] “Triangle inequality violations due to ixps, public page.” <http://j.mp/mPyLMV>.
- [69] “Team cymru ip to asn mapping.” <http://www.team-cymru.org/Services/ip-to-asn.html>.
- [70] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee, “Triangle inequality variations in the internet,” in *Internet Measurement Conference*, pp. 177–183, 2009.
- [71] “Nvidia corporation: Nvida cuda compute unified device architecture programming.” <http://developer.nvidia.com/cuda-toolkit-40>.
- [72] M. Ahmad and R. Guha, “Analysis of large scale traceroute datasets in internet routing overlays by parallel computation,” *The Journal of Supercomputing*, vol. 62, pp. 1425–1450, 2012. 10.1007/s11227-012-0811-9.
- [73] C.-H. Lin, S.-Y. Tsai, C.-H. Liu, S.-C. Chang, and J.-M. Shyu, “Accelerating string matching using multi-threaded algorithm on gpu,” in *GLOBECOM*, pp. 1–5, 2010.
- [74] M. Z. Ahmad and R. Guha, “Studying the effect of internet exchange points on internet link delays,” in *In proceedings of the Communications and Networking Symposium (CNS), SCS SpringSim 2010*, (Orlando, FL), April 2010.

- [75] M. Ahmad and R. Guha, “Evaluating end-user network benefits of peering with path latencies,” in *Proceedings of International Conference on Computer Communication Networks (ICCCN)*, July 2012.
- [76] “Caida: As relationships.” <http://www.caida.org/data/active/as-relationships/>.
- [77] L. Gao, “On inferring autonomous system relationships in the internet,” *IEEE/ACM Trans. Netw.*, vol. 9, pp. 733–745, December 2001.
- [78] “Networkx python language data structures for graphs.” Available at <http://networkx.lanl.gov/>.
- [79] M. Ahmad and R. Guha, “A tale of nice internet exchange points: Studying path latencies through major regional ixps,” in *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, oct. 2012.
- [80] “Generalized linear model.” http://en.wikipedia.org/wiki/Generalized_linear_model.
- [81] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2, IJCAI’95*, (San Francisco, CA, USA), pp. 1137–1143, Morgan Kaufmann Publishers Inc., 1995.