PERFORMANCE ANALYSIS AND PRIVACY PROTECTION OF NETWORK DATA

By

Faraz Ahmed

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

Computer Science—Doctor of Philosophy

2018

ABSTRACT

PERFORMANCE ANALYSIS AND PRIVACY PROTECTION OF NETWORK DATA

By

Faraz Ahmed

The goal of this thesis is to address network management research challenges faced by operational networks - with specific focus on cellular networks, content delivery networks, and online social networks. Next, I give an overview of my research on network management of these networks.

Cellular networks utilize existing service quality management systems for detecting performance degradation issues inside the network, however, under certain conditions degradation in End-to-End (E2E) performance may go undetected. These conditions may arise due to problems in the mobile device hardware, smartphone applications, and content providers. In this thesis, I present a system for detecting and localizing E2E performance degradation at cellular service providers across four administrative domains: cellular network, content providers, device manufacturers, and smartphone applications.

Cellular networks also need systems that can prioritize performance degradation issues according to the number of customers impacted. Cell tower outages are performance degradation issues that directly impact connectivity of cellular network users. In this thesis, we design and evaluate a cell tower outage monitoring system that analyzes and estimates device level impact during cell tower outages.

Content delivery networks (CDNs) maintain multiple transit routes from content distribution servers to eyeball ISP networks which provide Internet connectivity to end users. Two major considerations for CDNs are transit prices and performance dynamics of delivering content to end users. The dynamic nature of transit pricing and performance makes it challenging to optimize the cost and performance tradeoff. There are thousands of eyeball ISPs which are reachable via different transit routes and different geographical locations. Each choice of transit route for a particular eyeball ISP and geographical location has distinct cost and performance characteristics, which makes the problem of developing a transit routing strategy challenging. In this thesis, I present a measurement approach to actively collect client perceived network performance and then use these measurements towards optimal transit route selection for CDNs.

Online Social Networks (OSNs) often refuse to publish their social network graphs due to privacy concerns. Differential privacy has been the widely accepted criteria for privacy preserving data publishing. In this thesis, I present a random matrix approach to OSN graph publishing, which achieves storage and computational efficiency by reducing dimensions of adjacency matrices and achieves differential privacy by adding a small amount of noise. Dedicated to all those who value selflessness, friendship, and sacrifice; patience, tolerance, kindness, and compassion; perseverance, prudence, and hope; and integrity, honesty, humility, gratitude, and contentment.

ACKNOWLEDGEMENTS

Working towards a Ph.D. has been a deeply enriching and rewarding experience. Looking back, many people have helped shape my journey. I would like to extend them my thanks.

- First and foremost, my advisor, Prof. Alex X. Liu. My work would not have been possible without his constant guidance, his unwavering encouragement, his many insights, and his exceptional resourcefulness. And most importantly, his friendship. I have been very fortunate to have an advisor who has also been a close friend. For all of this, Alex, thank you.
- I would also like to thank the rest of my thesis committee Profs. Pang Ning Tan, Richard Lawrence Wash, and Jiayu Zhou for their encouragement and insightful comments during my comprehensive exams.
- I would also like to thank Dr. Zubair Shafiq. I learnt a lot from him. My collaboration with him has been one of the most fruitful and fun engagements I have experienced.
- I would also like to thank Dr. Charles B. Owen. Working as his Teaching Assistant has been one of my most valuable teaching experience. I learned a lot and had a lot of fun.
- I would also like to thank Drs. Jia Wang, Zihui Ge, Jeffrey Erman, and He Yan. I really enjoyed working with them during my summer internship at AT&T Labs - Research.
- Throughout my Ph.D., I was supported by various NSF research grants. Thanks NSF!
- I would also like to thank Michigan State University, and specifically Department of Computer Science and Engineering for providing me financial support to attend various conferences during my Ph.D.
- Many thanks to my colleagues in Systems and Security Lab at Michigan State University. In particular, I would like to thank Ali Munir, Ann Wang, Kamran Ali, Xinyu Lei, and Salman Ali for numerous insightful and inspirational discussions on various projects.
- I would like to thank my mentors, colleagues, and friends who helped me navigate life

as a Ph.D. student. In particular, I would like to thank Drs. Hassan Aqeel Khan, Muhammad Shahzad, Muhammad Zeeshan, and Mudassir Shabbir.

- I must say that I owe my great time in Michigan State University to all of my fabulous friends. It is simply not feasible to list all of them here. I would like to thank them all for their friendship and support.
- I am also very thankful to Drs. Muhammad Abulaish and Muddassar Farooq, who advised me before my Ph.D. and encouraged me to pursue Ph.D.
- I am also very thankful to these great personalities from my high school in my hometown Bahawalpur, Pakistan; Mr. Saleem, Mr. Rauf, and Mrs. Khalida Sarfraz. Looking back at the trajectory of my professional pursuits, I always find them at its origin and as its cause.
- Finally, I do not know how I can thank my family enough. First and foremost my parents, from whom I realized that kindness and devotion is endless, my siblings, who have always supported me, my wife, and the rest of the family members who have been a source of happiness.

TABLE OF CONTENTS

\mathbf{LI}	ST (OF TA	BLES	i
\mathbf{LI}	ST (OF FIC	GURES	ii
\mathbf{LI}	ST (OF AL	GORITHMS	v
1	Intr 1.1	oducti Contri 1.1.1	on	1 2
		1.1.2	works	2
			Cellular Networks	3
		1.1.3	Optimizing Transit Route Selection for CDNs	3
		1.1.4	Measurement and Analysis of Adult Traffic	4
		1.1.5	Privacy Preserving Social Network Graph Data Publishing	5
	1.2	Publis	hed Material	5
2	Perf	forman	he Degradation Detection and Localization in Cellular Networks $$	7
	2.1	Introd	uction \ldots	7
		2.1.1	Background and Motivation	7
		2.1.2	Proposed Approach	0
		2.1.3	Technical Challenges and Solutions	1
	2.2	Relate	d Work $\ldots \ldots 1$	2
	2.3	Monite	oring E2E Performance Dynamics and Data Analysis	4
		2.3.1	Data Collection	4
		2.3.2	Data Analysis	6
	2.4	Model	ing E2E Performance Dynamics	1
		2.4.1	Aggregate E2E Performance Matrix	2
		2.4.2	Coarse Grained E2E Modeling	3
		2.4.3	Fine Grained E2E Modeling 24	5
			2.4.3.1 Deviating E2E Instance Identification	7
			2.4.3.2 Deviating E2E Instance Grouping	7
			2.4.3.3 Recursive E2E Instance Grouping	0
	2.5	E2E P	Performance Degradation	1
		2.5.1	Detection	1
		2.5.2	Localization	1
	2.6	Evalua	ation \ldots \ldots \ldots \ldots 3°	2
		2.6.1	Synthetic Anomalies	2

		2.6.2	Anomaly Detection in the Wild	35
		2.6.3	Real-World Anomalies	38
	2.7	Conclu	isions	39
3	Ass	essing	Impact of Cell Tower Outages in Cellular Networks	40
	3.1	Introd	\mathbf{u} ction	40
		3.1.1	Background and Motivation	40
		3.1.2	Limitations of Prior Art	41
		3.1.3	Proposed Approach	42
		3.1.4	Technical Challenges and Solutions	43
		3.1.5	Key Novelty and Advantages	45
	3.2	Relate	d Work	45
	3.3	Backgr	round and Data Collection	46
		3.3.1	Dataset	47
	3.4	Spatio	-Temporal Characterization	49
	3.5	System	n Design	52
		3.5.1	Activity Profiling	52
		3.5.2	Location Profiling	53
		3.5.3	Quantifying Temporal Impact	53
		3.5.4	Quantifying Spatial Impact	54
	3.6	Evalua	ution	56
		3.6.1	Synthetic Network Outages	57
		3.6.2	Real Network Outages	60
	3.7	Conclu	sion	63
4	Opt	imizing	g Transit Route Selection for CDNs	64
	4.1	Introd	uction	64
	4.2	Backgr	cound	67
		4.2.1	Architecture	67
		4.2.2	Internet Transit Dynamics	69
	4.3	Perform	mance Measurements & Analysis	70
		4.3.1	Measurement Methodology	70
		4.3.2	Analysis and Discussions	73
	4.4	Proble	m Formulation	77
	4.5	Result	S	81
		4.5.1	Tradeoff analysis	83
		4.5.2	Transit selection analysis	84
	4.6	Pareto	Front Generation \cdot	89
	4.7	Relate	d Work	92
	4.8	Conclu	nsions	93

5	Mea	asurement and Analysis of Adult Traffic	5
	5.1	Introduction	5
		5.1.1 Background $\ldots \ldots $	5
		5.1.2 Limitation of Prior Art	6
		5.1.3 Main Findings $\dots \dots \dots$	6
	5.2	Related Work	8
	5.3	Measurement & Analysis	9
		5.3.1 Aggregate Analysis	1
		5.3.1.1 Content Composition $\ldots \ldots \ldots$	1
		5.3.1.2 Traffic Composition $\ldots \ldots \ldots$	1
		5.3.1.3 Temporal Access Patterns $\ldots \ldots \ldots$	3
		5.3.1.4 Device/OS Usage $\ldots \ldots \ldots$	3
		5.3.2 Content Dynamics $\ldots \ldots \ldots$	4
		5.3.2.1 Content Size $\ldots \ldots \ldots$	4
		5.3.2.2 Content Popularity $\ldots \ldots \ldots$	5
		5.3.2.3 Impact of Content Injection and Aging on Popularity \ldots 10'	7
		5.3.3 User Dynamics $\ldots \ldots \ldots$	1
		5.3.3.1 User Request Inter-arrival Time	1
		5.3.3.2 User Session Length $\ldots \ldots \ldots$	1
		5.3.3.3 User Addiction $\ldots \ldots \ldots$	3
	5.4	Implications $\ldots \ldots \ldots$	4
		5.4.1 CDN Cache Hit Ratios $\ldots \ldots \ldots$	5
		5.4.2 HTTP Response Codes	6
	5.5	$Conclusion \dots \dots$	6
6	Priv	vacy Preserving Social Network Graph Data Publishing	3
	6.1	Introduction $\ldots \ldots \ldots$	8
		6.1.1 Background and Motivation	8
		6.1.2 Problem Statement	9
		6.1.3 Limitations of Prior Art	0
		6.1.4 Proposed Approach	0
		6.1.5 Technical Challenges	1
		6.1.6 Key Contributions	1
	6.2	Related Work $\ldots \ldots 122$	2
	6.3	Random Matrix Approach	3
		6.3.1 Theoretical Guarantee on Differential Privacy	5
		$6.3.2 \text{Proof of Theorem 1} \dots $	6
		6.3.3 Theoretical Guarantee on Eigenvector Approximation	7
		$6.3.4 \text{Proof of Theorem } 2 \dots \dots$	8
	6.4	Experimental Results	1
		$6.4.1$ Dataset \ldots 132	2

		6.4.2	Node	Cluste	ering	g.		•	 	•	•						 •			•					133
		6.4.3	Node	Ranki	ng .				 	•	•			•		•	 •		•	•		•	•		138
	6.5	Conclu	isions	•••	•••			•	 	•				•		•	 •	•	•	•		•	•	•	140
8	Con	clusior	1			•	•••	•	 •		•	•	 •	•••	•	•	 •	•	•	•	•		•	•	141
BIBLIOGRAPHY																									

LIST OF TABLES

Table 2.1	RMSE values for different weighting functions and regression methods	22
Table 2.2	Single dimension anomalies	36
Table 2.3	Multi-dimension anomalies	36
Table 4.1	Data set summary statistics	73
Table 4.2	Summary of assignments for different pricing rates and POP locations	85
Table 6.1	Dataset description	133
Table 6.2	Memory utilization and running time for the proposed algorithm $\ . \ .$	137
Table 6.3	NMI for clustering using LNPP Approach [141] for $\sigma = 1$	138
Table 6.4	$n \times \text{MSE}$ using the proposed approach	140
Table 6.5	$n \times MSE$ using baseline LNPP	140

LIST OF FIGURES

Figure 2.1	Normalized TCP Loss Ratio	14
Figure 2.2	Normalized Round Trip Time (RTT)	14
Figure 2.3	Cellular network architecture	15
Figure 2.4	CDF of Round Trip Time (RTT) in 4G LTE networks	19
Figure 2.5	Heatmap plots for combination of dimensions	21
Figure 2.6	Residual error distribution	25
Figure 2.7	Real and predicted performance curves	26
Figure 2.8	No. of outliers	26
Figure 2.9	Single model	26
Figure 2.10	Multiple models	26
Figure 2.11	Synthetic anomaly detection	33
Figure 2.12	Real-world anomaly 1: root cause outside the cellular network $\ . \ . \ .$	37
Figure 2.13	Real-world anomaly 2: root cause inside the cellular network \hdots	37
Figure 3.1	Spatio-temporal characterization of M2M devices	49
Figure 3.2	Active device density	51
Figure 3.3	Overview of system architecture	52
Figure 3.4	Example devices	53
Figure 3.5	Estimation accuracy of start times and duration of synthetic outages	56
Figure 3.6	Estimation accuracy of user impact in synthetic outages	58
Figure 3.7	Average of absolute difference in timings	59
Figure 3.8	Device impact of real outages	59
Figure 3.9	Heatmap visualization of real outage for different p-value threshold .	60
Figure 3.10	A special case of real world site outage	60
Figure 4.1	A CDN interconnects with multiple transit ISPs at IXPs $\ . \ . \ .$.	68
Figure 4.2	Transit performance measurements	71

Figure 4.3	Deployment summary	73
Figure 4.4	Spatial performance variations for different POPs	75
Figure 4.5	Performance difference of transit providers for different billing periods	76
Figure 4.6	Difference in transit performance for AS-POP pair: Telefonica-Madrid	77
Figure 4.7	Difference in transit performance for AS-POP pair: ${\tt Comcast-San}$ Jose	77
Figure 4.8	Cost-performance tradeoff curves for γ ranging from 0 to 5	82
Figure 4.9	Transit provider selection for different pricing scenarios	86
Figure 4.10	Transit provider selection for varying price ratio: $\frac{\text{TEL } price}{\text{NTT } price}$	87
Figure 4.11	Effect of γ values on transit provider selection (Paris)	88
Figure 4.12	Analysis of transit provider bandwidth usage	88
Figure 4.13	Pareto fronts for different hours	89
Figure 5.1	Content composition of five adult websites.	101
Figure 5.2	Traffic composition of five adult websites	102
Figure 5.3	Hourly traffic volume timeseries of five adult websites. \ldots	103
Figure 5.4	Device type composition	104
Figure 5.5	Content size distributions.	105
Figure 5.6	Content popularity distributions	106
Figure 5.7	Fraction of total object requested for adult websites at different ages .	107
Figure 5.8	Content clustering dendrograms of two adult websites. \ldots	107
Figure 5.9	Cluster medoids for V-2 adult website	108
Figure 5.10	Cluster medoids for P-2 adult website	108
Figure 5.11	User request inter-arrival time distributions	111
Figure 5.12	User session length distributions	112
Figure 5.13	Repeated access of objects	112
Figure 5.14	CDF of repeated content access by users	113
Figure 5.15	Hit ratio for image and video objects	114
Figure 5.16	Response codes	115

Figure 6.1	Degree distribution of three datasets	132
Figure 6.2	NMI values for Facebook	133
Figure 6.3	NMI values for Live Journal	133
Figure 6.4	NMI values for Pokec	134
Figure 6.5	Cluster distribution for Facebook	136
Figure 6.6	Cluster distribution for Live Journal	137
Figure 6.7	Cluster distribution for Pokec	137
Figure 6.8	Percentage of preserved ranks using random matrix approach	138

LIST OF ALGORITHMS

Algorithm 1	Estimating outage start time and duration	55
Algorithm 2	Generating pareto fronts recursively	91
Algorithm 3	$\widehat{A} = \operatorname{Publish}(A, m, \sigma^2)$	124

1 Introduction

Operational networks, such as Cellular Networks, Content Delivery Networks (CDNs), and Online Social Networks (OSNs) are rapidly changing in terms of network topology, networking infrastructure, and network traffic characteristics. Network operators frequently change peering agreements, routing policies, networking equipment, etc., to improve their network performance and end-user Quality of Experience (QoE). In addition, external entities like content providers, end-user device manufacturers and firmware developers also contribute to the constantly changing internet landscape. Upgrades in network infrastructure, new network protocols, new content types, etc., inadvertently creates new network management challenges for operational networks.

In this thesis, I address network management needs by measuring network performance, and modeling and analyzing network data for improving network performance. For cellular networks, I focus on modeling of network performance. I have used statistical and machine learning methods to develop system that solve network management issues involving detection and localization of network performance degradation. I also present my work on assessing the customer impact of cell tower outages. For content distribution networks (CDNs), I focus on developing approaches to actively collect client perceived network performance. I then use these measurements along with server side logs for modeling and analysis. I present approaches that solve research problems involving quantification and optimization of CDN performance. For online social networks, I present my work on developing privacy preserving approaches for graph data publishing.

1.1 Contributions

This thesis takes an in-depth look at the following research problems.

1.1.1 Performance Degradation Detection and Localization in Cellular Networks

Cellular network users can experience performance degradation that can occur due to issues outside the cellular network, it may be due to the user device itself (such as device OS, application software) or due to the content provider (such as application servers, datacenter network). When users experience degradation in performance, they call customer care representatives of the cellular network. When cellular service providers receive such calls, they have to go through lengthy and labor intensive processes to first verify the issue, and then solve it. Cellular networks utilize existing service quality management systems for detecting performance degradation issues inside the network, however, degradation in performance due to issues outside the network may go undetected. In this work, I build a comprehensive and holistic system that monitors E2E service performance across four dimensions - user locations, content providers, device types, and application types [23, 24, 17]. I model the expected service performance for each combination of service location, content provider, device type, and application type using training data, then compare ongoing E2E performance measurement with the model and detect performance degradation as it arises, and finally localize the service degradation using an associate rule mining based approach. The system detected several performance degradation instances over a period of one week in an operational cellular network. In 80% of the detected degraded instances, content providers, device types, and application types were the only factors of performance degradation. The evaluation using both real network traces and synthetically generated performance degradation shows that our method is highly effective.

1.1.2 Assessing Impact of Cell Tower Outages in Cellular Networks

A major network management challenge faced by network operators is to identify service disruptions such as cell tower outages, in a timely manner and assess impact of service disruptions on cellular network users. Typically, many service disruptions can occur concurrently in a cellular network. Network operators want to prioritize these outages according to the customer impact. Due to scalability constraints, only spatially and temporally sampled customer level measurements are collected, which are typically insufficient for cellular network operators to determine the customer impact. I design and evaluate a cell tower outage monitoring system that analyzes and estimates the device level impact during cell tower outages [18]. I propose to look at the network activity of Machine-to-Machine devices that communicate using the same cellular network as human operated devices such as mobile phones. We first identify a candidate set of devices with highly predictable mobility and traffic patterns using data collected from a large-scale cellular provider in North America. We then use these qualified M2M devices to observe the impact of cell tower outages from end users perspective. We evaluate the system by using cell tower outages in a 2-months time window and our operational experience reveals that: (i) network quality can be analyzed at a higher granularity by monitoring M2M device communication; (ii) customer impact during network outages varies both in space and time; and (iii) customer impact estimates can be used by cellular network provider to improve network coverage and network resiliency. To the best of our knowledge, this is the first work that employs M2M devices to assess the impact of service issues in operational cellular networks at a large scale.

1.1.3 Optimizing Transit Route Selection for CDNs

Two major considerations for CDNs are cost and performance of delivering content to end users. Pricing and performance of transit providers vary with respect to time and geographical location. Due to the dynamics of varying performance and pricing on transit routes, CDNs need to implement a transit route selection strategy to optimize performance and cost tradeoffs. CDNs have multi-homed servers at IXPs (Internet Exchange Point) which provide explicit control over content routing via multiple transit routes. There are thousands of eyeball ISPs which are reachable via different transit routes and different geographical locations. Each choice of transit route for a particular eyeball ISP and geographical location has distinct cost and performance characteristics. In this work, I first implement a client-side performance measurement JavaScript to obtain simultaneous performance measurements across multiple transit routes and then formalize the transit routing problem using a multi-attribute objective function to simultaneously optimize end-to-end performance and cost [22, 20]. The measurement methodology allows CDNs to simultaneously capture user perceived end-to-end performance via multiple transit routes. The optimization methodology allows CDNs to navigate the cost and performance tradeoff in transit routing. I evaluate the approach using real-world measurements from CDN servers located at 19 geographically distributed IXPs. The results show that CDNs can achieve significant cost and performance benefits using the measurement and optimization approach.

1.1.4 Measurement and Analysis of Adult Traffic

Adult (or pornographic) websites attract a large number of visitors and account for a substantial fraction of the global Internet traffic. However, little is known about the makeup and characteristics of online adult traffic. In this work, we present the first large-scale measurement study of online adult traffic using HTTP logs collected from a major commercial content delivery network [21]. We analyze several characteristics of online adult traffic including content and traffic composition, device type composition, temporal dynamics, content popularity, content injection, and user engagement. Our analysis reveals several unique characteristics of online adult traffic. We also analyze implications of our findings on adult content delivery. Our findings suggest several content delivery and cache performance optimizations for adult traffic, e.g., modifications to website design, content delivery, cache placement strategies, and cache storage configurations.

1.1.5 Privacy Preserving Social Network Graph Data Publishing

Online Social Networks (OSNs) often refuse to publish their social network graphs due to privacy concerns. Differential privacy has been the widely accepted criteria for privacy preserving data publishing. In this work, we propose a random matrix approach to OSN graph publishing, which achieves storage and computational efficiency by reducing dimensions of adjacency matrices and achieves differential privacy by adding a small amount of noise [19]. The key idea is to first project each row of an adjacency matrix into a low dimensional space using random projection, and then perturb the projected matrix with random noise, and finally publish the perturbed and projected matrix. We prove that random projection plus random perturbation preserve differential privacy, and also that the random noise required to achieve differential privacy is small. We validate the proposed approach and evaluate the utility of the published data for three different applications, namely node clustering, node ranking, and node classification, using publicly available OSN graphs of Facebook, Live Journal, and Pokec.

1.2 Published Material

The chapters of this dissertation are based in part on the following publications.

- Faraz Ahmed, Zubair Shafiq, Amir Khakpour, and Alex X. Liu. "Optimizing Internet Transit Routing for Content Delivery Networks", IEEE/ACM Transactions on Networking (ToN), 2018.
- Faraz Ahmed, Jeffrey Erman, Zihui Ge, Alex X. Liu, Jia Wang, and He Yan. "Detecting and Localizing End-to-End Performance Degradation for Cellular Data Services Based on TCP Loss Ratio and Round Trip Time", IEEE/ACM Transactions on Networking

(ToN), 2017.

- Faraz Ahmed, Jeffrey Erman, Zihui Ge, Alex X. Liu, Jia Wang, and He Yan. "Monitoring Quality-of-Experience for Operational Cellular Networks Using Machine-to-Machine Traffic", IEEE INFOCOM, 2017
- Faraz Ahmed, Zubair Shafiq, Amir Khakpour, and Alex X. Liu. "Optimizing Internet Transit Routing for Content Delivery Networks", IEEE **ICNP**, 2016.
- Faraz Ahmed, Jeffrey Erman, Zihui Ge, Alex X. Liu, Jia Wang, and He Yan. "Detecting and Localizing End-to-End Performance Degradation for Cellular Data Services", IEEE INFOCOM 2016.
- Faraz Ahmed, Alex X. Liu, and Rong Jin. "Social Graph Publishing with Privacy Guarantees", IEEE ICDCS 2016.
- Faraz Ahmed, M. Zubair Shafiq, and Alex X. Liu. "The Internet is for Porn: Measurement and Analysis of Online Adult Traffic", IEEE ICDCS 2016.
- Faraz Ahmed, Jeffrey Erman, Zihui Ge, Alex X. Liu, Jia Wang, and He Yan. "Detecting and Localizing End-to-End Performance Degradation for Cellular Data Services", IEEE
 SIGMETRICS [Extended Abstract] 2015.

2 Performance Degradation Detection and Localization in Cellular Networks

2.1 Introduction

2.1.1 Background and Motivation

Internet access through cellular data services has become an essential part of people's everyday life such as emailing, web browsing, video streaming, and online shopping. Nowadays cellular network customers using data services expect seamless service with high application performance, i.e., the end-to-end (E2E) performance. To best serve their customers it is critical for cellular service providers to provide high E2E performance and maintain their competitive edge. In the context of cellular data services, E2E performance means the performance that customers experience for a specific location, content provider, device type, and application type. A *user location* means the Radio Network Controller (RNC) that the user's device connects to. A *content provider* means the Internet domain that is serving the user. A *device type* means a specific brand and model of the user's device, such as Apple iPhone 5. An *application type* means the categorical type of the application that the user is running on their cellular device, such as emailing or Web browsing. Prior work has shown that E2E performance in cellular networks varies across the above mentioned attributes [27],[72]. Along with the cellular network's performance, the hardware and software performance of different device types and applications also largely define the E2E performance over the cellular network. A group of users may experience bad service due to problems in their device firmware, applications and/or due to problems at the network side. For example, degradation in performance can be due to issues such as limited processing power of smartphones devices in combination with device operating systems and applications running on them [73]. Thus, the inherently complex nature of cellular data services creates challenges for cellular data service providers to manage *Quality of Experience* (QoE) from the subscribers' point of view.

In this work, we present a system for detecting and localizing E2E performance degradation (such as slow webpage loading and unsmooth video playing) at cellular service providers across four administrative domains: cellular service providers, content providers, device manufacturers, and application developers. For detection, we want to detect E2E performance degradation before cellular network operators receive complaint calls. For localization, we want to find the problematic domain that is causing performance degradation such as user location, content provider, device type, and/or application type. For example, if all users connecting to an RNC are experiencing performance degradation, regardless of content providers, device types, and application types, then probably the RNC is causing the performance degradation. For another example, if all users of iPhone 5 are experiencing performance degradation for their email application, regardless of user locations and content providers, then probably iPhone 5 is having issues with email applications.

Detection and localization of E2E performance degradation is crucial for all administrative domains to jointly troubleshoot root causes. When users experience E2E performance degradation, they have no clue whom they should "blame". For example, when a user at New York experience performance degradation for yahoo email on his iPhone 5, he does not know whether it is the cellular provider, or yahoo, or iPhone 5, or his email client app, that is causing the problem. When user experience such E2E performance degradations, they blankly ascribe the fault to their cellular service providers and issue complaints to the cellular network customer service centers, which may result in both reputation damage and financial losses for the cellular service providers. When cellular service providers receive such calls, they have to go through lengthy, manual, labor intensive process to localize the issue, and the actual issue may not be the cellular service provider problem, it may be due to the user device itself (such as device OS, application software) or due to the content provider (such as application servers, datacenter network). For example, the E2E performance degradation maybe caused by a content provider upgrading its service, a device type having updated its OS with incompatibility issues, and an app having been patched with buggy code. The localization findings allow cellular service providers to quickly mitigate service problems, effectively communicate with customers complaining about service problems, and engage content providers, device manufactures, or application developers and operators to jointly troubleshoot for root causes. They sometimes can walk around the E2E performance issues on their own. For example, if the issue is caused by a neighboring network, then they may adjust their intra/inter domain routing to avoid that network.

There are several key challenges in detecting and localizing E2E performance degradation at cellular service providers. First, there are a wide range of elements (such as mobile devices, cell towers, radio resource controller, routers, switches, fibers, media gateways, firewalls, multicast servers, name servers, and content servers) at various layers (such as physical layer, link layer, transport layer and application layer) that may cause the E2E performance degradation. Furthermore, the heterogeneity of mobile devices increases the number of factors contributing to the E2E performance degradation. For example, there are over 24k distinct device types from Android. Over 2 million Android applications access content from millions of content providers through cellular networks [110]. Similarly, other device manufacturers such as Windows, Apple etc have numerous device models. Applications developed for different operating systems running on these devices increase the dimensionality.

Second, it is practically infeasible to gain the complete visibility of E2E performance

issues because content providers, devices, and applications belong to different administrative domains. For example, issues on mobile devices and content servers are not visible to cellular service providers. Third, the expected E2E performance varies significantly depending on which application (such as web browsing, video streaming, or gaming) requesting information from which content provider (such as youtube.com or google.com) running on which mobile device (such as Apple iPhone 5 or Motorola V3 Razr) from which geographic location (such as New Jersey suburban areas or NYC downtown areas) at what time of day and during which day of a week [71, 72, 125].

2.1.2 Proposed Approach

In this chapter, we propose a holistic approach to detecting and localizing E2E performance degradation at cellular service providers across the four administrative domains of user locations, content providers, device types, and application types. Our approach consists of three steps: modeling, detection, and localization.

First, we build models using training data to capture the normal performance of individual *E2E instance*, which means the flows corresponding to a particular user location, content provider, device type, and application type. Each E2E instance has 24 * 7 models where each model corresponds to a specific hour of a day and a specific day of a week. We build different models for each hour in a day and each day in a week because studies have shown that people have distinct daily and weekly diurnal patterns in their use of cellular services [125]. Our training data, which is collected from a major cellular service provider in North America, contains anonymized TCP flow level information such as standard coordinated universal time (UTC), the serving RNC, the type of the cellular device, the type of the application, and the Internet domain that is being accessed.

Second, we use our models to detect performance degradation for each E2E instance on hourly basis. For each E2E instance, if the actual performance in the testing phase is worse than the expected performance obtained through our models, we label it as degrading. Third, after marking each E2E instance non-degrading or degrading, we use association rule mining techniques to localize the source of performance degradation. For example, rule iPhone 5, Email \rightarrow degrading shows that at a particular time instant, for all locations and content provides, the cellular users of iPhone 5 are experiencing significant degradation in performance when they use the email application.

2.1.3 Technical Challenges and Solutions

To implement our approach, we face three key technical challenges. The first challenge is to tradeoff between model accuracy and model complexity and to deal with data sparsity. We represent E2E performance in a four dimensional matrix that is called *E2E matrix* and is denoted $E_I = [1..L, 1..P, 1..D, 1..A]$, where the four dimensions are L user locations, P content providers, D device types, and A application types. An element $E_I[l, p, d, a]$ in this matrix represents the E2E instances corresponding to user locations l, content provider p, device type d, and application a. On one extreme end, we build only one model for all E2E instances in the E2E matrix, which gives us the least accuracy and the least complexity; on the other extreme end, we build one model for each individual E2E instance, which in theory gives us the most accuracy and the most complexity. Actually for many E2E instances, we do not have enough flow data to build meaningful models. To deal with data sparsity we first aggregate all flows belonging to same E2E instance. We use the aggregated flows to obtain a two-dimensional aggregate E2E performance matrix denoted as E_A . The E_A matrix is our design matrix used as input to the regression algorithm. To deal with the tradeoff between model accuracy and complexity, use the aggregate E2E matrix E_A to build a baseline model for all E2E instances, identify the E2E instance groups that have significantly different performance, and then model these groups separately, leaving the rest E2E instances still being modeled by the baseline model. Within each group, the performance of some E2E instances may be relatively different from that of others in the group; thus, we apply this grouping strategy recursively among identified groups.

The second challenge is to localize E2E performance degradation issues. After we label each E2E instance as either non-degrading or degrading, we cannot directly present such a labeled E2E matrix to network operators because the number of degrading E2E instances maybe too large. To address this challenge, we use association rule mining to summarize the E2E matrix using some simple rules such as iPhone 5, Email \rightarrow degrading.

The third challenge is to quantitatively evaluate the effectiveness of our approach because we are short of ground truth data. For detection, we do not have pre-labeled testing data because labeling such prohibitive amount of data is practically infeasible. For localization, the customer tickets that we have access to mostly document hard failures such as an RNC is down. When such hard failures happen, typically there is no flow for the affected E2E instances; therefore, our localization scheme will not find such failures. To address this challenge, first, we performed manual inspection for some E2E performance degradation cases; second, we injected some synthetic performance degradation cases into the test data and use those injected cases to serve as the ground truth.

2.2 Related Work

To the best of our knowledge, the detection and localization of E2E performance degradations at cellular service providers has not been previously studied. Effort related to ours can be categorized into detecting and troubleshooting network or service issues and cellular service E2E performance measurements.

Detecting and Troubleshooting Network or Service Issues: SCORE [89], Shrink [80], and [90] focus on diagnosing network failures through the inference of underlying lowerlayer failures using a Shared Risk Link Group (SRLG) model. Pinpoint focuses on diagnosing the root causes of service failures at the application server end, *i.e.*, the content provider end [41]. Sherlock [32] and NetMedic [81] both focus on diagnosing the root causes of service failures within the elements (such as DNS servers, firewall configurations, and application servers) of an enterprise network and both use dependency graphs. Sherlock and NetMedic conduct diagnosis at the granularity of machines and processes, respectively. In [98], Mahimkar *et al.* focused on characterizing and troubleshooting performance issues in one of the largest IPTV networks in North America. In [99], Mahimkar *et al.* designed and implemented a tool for detecting the impact of network upgrades on performance. Our work is different from these efforts in terms of both the problem being solved and the solution being used. From the problem perspective, our problem is on monitoring, detecting, and localizing E2E performance degradations for cellular data service users along the four dimensions of user locations, content providers, device types, and application types at cellular service providers without the full visibility of the E2E service path. From the solution perspective, our approach consists of passive cellular traffic monitoring, data centric modeling, model based performance degradation detection, and association rule based root cause localization, whereas these efforts uses variations of dependency graphs.

Cellular service E2E performance Measurements: There is work on measuring the impact on cellular service E2E performance by factors such as resource usage by mobile applications [114], content providers [50], and cellular technologies [72]. In [114], Qian *et al.* designed and implemented a tool for discovering inefficient resource usage for smartphone applications using cross-layer interaction among various layers including radio resource channel state, transport layer, application layer, and the user interaction layer. In [50], Finamore *et al.* performed a measurement study that compares YouTube traffic generated by mobile devices with that generated by PCs, and correlated user behavior with system performance. In [72], Huang *et al.* studied the interactions among applications, network transport protocols, and radio layers for the 4G LTE technology and their impact on performance using both active and passive measurements, and showed that E2E performance (such as TCP loss ratio) in cellular services varies across different content providers, device types, applications, connection types, and wireless carriers. Comparing these efforts with ours, we go one step further to localize the causes of E2E performance degradation. The closest work to ours is



Figure 2.2 Normalized Round Trip Time (RTT)

presented in [27]. The authors look at aggregated Round Trip Time (RTT) measurements in the UMTS network to identify poorly performing attributes including RNC, device, application and content provider. In comparison, we look at two metrics RTT and TCP loss ratio. We present a comprehensive approach that includes measurement and modeling to detect and localize performance degradation on hourly basis. In addition to real world case studies, we evaluate our approach by detecting and localizing anomalies in the wild over a period of one week.

2.3 Monitoring E2E Performance Dynamics and Data Analysis

2.3.1 Data Collection

In this study, we utilize anonymized flow level data collected from the core network of a major cellular service provider in the United States. Figure 2.3 illustrate the architectural overview



Figure 2.3 Cellular network architecture

of the core of the mobile network, which consists of two main types of nodes: the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN). The GGSN is the root node in the hierarchy of the cellular data network. GGSN is responsible for sending and receiving Internet traffic to and from the cellular network. SGSN is an intermediate node that connects the lower level nodes to the GGSN through the G_n interface. Typically, a single SGSN is connected to multiple Radio Network Controllers (RNCs) and each RNC serves a geographical region through cell towers.

The data was collected at the *Gn* interface which connects the SGSNs to the GGSNs. The data collection system logs TCP flow level information for each user equipment. Therefore, our system takes as input the incoming data feed and outputs the detection and localization results. As our goal in this work is to detect and localize the E2E performance degradation in the cellular service, we collect TCP flow level information for each TCP connection which has been aggregated by user equipment (UE) or handheld device type, the particular RNC, SGSN, and GGSN.

Each aggregated record in addition contains the timestamp of the (1hr) bin, the application type (e.g. web browsing, streaming video, etc) and content provider (e.g. www.something.com). In order to deal with data sparsity, we limit our time granularity to 1hr time bins. This limits our system to detect anomalies on hourly basis. For each TCP

flow, we collect information including standard coordinated universal time (UTC), ID of the serving RNC that describes the user access point, the device type, the application type, and the content provider being accessed. In this work, we focus on two most important E2E performance metrics in a cellular service: TCP loss ratio and Round Trip Time (RTT). For each flow, we calculate its TCP loss ratio using the following formula:

$$\left(rac{ ext{observed }\# ext{ bytes in the flow }}{ ext{actual }\# ext{ bytes in the flow }}-1
ight)$$

Here the observed number of bytes in a flow means the number of bytes that is transmitted in the network for the flow including those being retransmitted due to packet loss. The actual number of bytes in a flow means the total number of bytes in the flow, excluding retransmissions. We detect retransmitted packets by tracking packet sequence numbers. In this formula, the observed number of bytes is equal to the actual number of bytes in the flow plus the number of retransmitted bytes. For each flow, we calculate its RTT using two RTT measurements. The first measurement corresponds to the cellular network side of the Gninterface. The second measurement corresponds to the internet side of the Gn interface. The E2E RTT of a flow is equal to the sum of cellular network side RTT and internet side RTT. Note that all user/device identifiers (such as IMSI and IMEI) are completely anonymized to protect user privacy.

2.3.2 Data Analysis

To understand the characterization of TCP Loss Ratio and RTT across the four dimensions (namely user location, content provider, device type, and application type), we examine one week of data collected from one northeast region in the United States. We now analyze the dynamics in TCP loss ratio and RTT for each of the 4 dimensions. The insights that we gain in such analysis will be useful in our modeling of E2E performance.

User Location: Figure 2.1(a) shows the aggregate hourly TCP loss ratios for all user locations. We use RNC ID as an identifier for user location. Each RNC connects hundreds

of cell towers to the core network, covering a large geographical region. Figure 2.1(a) plots TCP loss ratios of 78 different RNCs covering multiple states in the United States. Here each grey line represents aggregate hourly TCP loss ratio of a single RNC and the black line represents the average. The *aggregate hourly TCP loss ratio of a user location* is calculated based on the sum of the observed number of bytes transmitted for all downlink flows divided by the sum of the actual number of bytes in all downlink flows using the following formula:

$$\left(\frac{\sum_{f \in \{\texttt{downlink flows}\}} \texttt{observed} \ \# \ \texttt{bytes in} \ f}{\sum_{f \in \{\texttt{downlink flows}\}} \texttt{ actual} \ \# \ \texttt{bytes in} \ f} - 1\right)$$

From this figure, we first observe that the aggregate TCP loss ratios of all user locations follow a similar diurnal pattern. Second, we observe that the aggregate hourly TCP loss ratios for all user locations demonstrate moderate deviation from their average. Figure 2.2(a) shows the aggregate hourly RTT for all user locations. From this figure, we observe that RTTs of all user locations have moderate deviations from their average. This is because aggregate end-to-end RTT is dominated by the wireless portion of the cellular network [27]. Therefore, there is little deviation in aggregate RTTs across locations.

Content Provider: Figure 2.1(b) shows the aggregate hourly TCP loss ratios for top 20 popular content providers, where the grey lines represent the individual ratios and the black line represents the average. The *aggregate hourly TCP loss ratio of a content provider* is calculated in a similar fashion as that of a user location, except that the aggregation is for each content provider. From this figure, we first observe that the aggregate TCP loss ratios of these content providers follow a similar diurnal pattern as those of user locations. Second, we observe that the aggregate hourly TCP loss ratios of these top 20 content providers demonstrate significant deviation from their average, as compared to those ratios of user locations. Third, we observe that a few content providers constantly perform significantly worse than others. Figure 2.2(b) shows the aggregate hourly RTT values for the top 20 popular content providers. From this figure, we observe that some content providers demonstrate different RTT values over time as compared to their average. Also, similar to TCP loss ratio, RTT values of some content providers are significantly worse than others. The variations of TCP

loss ratio and RTT across different content providers show that E2E performance can be affected by the content provider. This could be because of several reasons such as content provider popularity, content type (e.g., video, image, text etc), different content delivery strategies which may not be optimized for mobile platform etc.

Device Type: Figure 2.1(c) shows the aggregate hourly TCP loss ratios for top 10 popular handheld cellular device types, where the grey lines represent the individual ratios and the black line represents the average. The *aggregate hourly TCP loss ratio of a device type* is calculated in a similar fashion as that of a user location, except that the aggregation is for each device type. From this figure, we first observe that the aggregate TCP loss ratios of these device types follow a similar diurnal pattern as those of user locations. Second, we observe that the aggregate hourly TCP loss ratios of these content providers demonstrate moderate deviation from their average, slightly higher than those ratios of user locations. Third, we observe that a few device types constantly perform moderately worse than others. Figure 2.2(c) shows the aggregate hourly RTT values for top 10 popular handheld cellular device types. From this figure, we observe that some device types demonstrate different RTT values over time as compared to their average. The variations can be associated to different hardware capabilities such as different screen resolutions, processing power, and other differences in mobile device hardware.

Application Type: Figure 2.1(d) shows the aggregate hourly TCP loss ratios for top 10 popular application types, where the grey lines represent the individual ratios and the black line represents the average. The *aggregate hourly TCP loss ratio of an application type* is calculated in a similar fashion as that of a user location, except that the aggregate TCP loss ratios of these application type. From this figure, we first observe that the aggregate TCP loss ratios of these application types follow a similar diurnal pattern as those of user locations. Second, we observe that the aggregate hourly TCP loss ratios of these top 10 application types demonstrate significant deviation from their average, as compared to those ratios of user locations. Third, we observe that a application type, which is email, constantly perform



Figure 2.4 CDF of Round Trip Time (RTT) in 4G LTE networks.

significantly worse than others. Figure 2.2(d) shows the aggregate hourly RTT values for top 10 popular application types,. From this figure, we observe that some applications demonstrate different RTT values over time as compared to their average. Also, similar to TCP loss ratio, RTT values of some applications are significantly worse than others. Variations in application performance can be attributed to resource usage and communication behavior of individual applications[114].

These performance differences also manifest in 4G LTE networks. Our data is collected on a 3G network, to verify the applicability of these findings in 4G LTE networks we utilize publicly available Google mobile networks measurements dataset [106]. The dataset covers 295 locations, 75 operating system versions, and 33 device types. The data is collected from 144 carriers and 9 network technologies all over the world. It contains 6.6 million RTT measurements. Due to limited information available in the data, we look at performance differences across different locations, device types and operating system versions. For, simplicity we only show top 10 locations, devices, and operating systems. Figure 2.4 shows the Cumulative Distribution Function of RTT measurements. Each line in the figure correspond to a single location, device type or an operating system. Figure 2.4(a) shows that performance differences across the top 10 locations. In comparison, we observe large performance differences across device types and operating system versions in 2.4(b) and (c) respectively. These observations show that irrespective of the cellular network technology, device hardware, content providers, and smartphone applications affect E2E performance.

We have analyzed the temporal dynamics of TCP loss rates and RTT by aggregating performance along each individual dimension (user locations, content providers, device types, or application types). It is also useful to understand E2E performance differences that manifest due to the interplay of any of these dimensions. To understand the relationships between different combinations of dimensions, we obtain average TCP loss rates for all combinations of two dimensions. For each combination of dimension, we obtain an aggregate performance matrix, where each row represent one attribute of a dimension (e.g., a device in Device Types) and each column of the matrix represent one attribute of another dimension (e.g., an application in Application Types). Each element of the matrix contains average E2E performance over a period of one week. The average is obtained by selecting all TCP flows identified by the two attributes. We provide a visualization of these matrices in Figure 3.9 that shows the heat map of different combinations of dimensions. The values of the matrix are normalized and the color bar shows the range of values in the heat map. Figure 3.9(a) shows the heat map of 50 locations and 25 devices. The y-axis represents locations and the x-axis represents devices. Each pixel in the figure represent the average TCP loss rate of a particular device type at a location. From this figure we observe that majority of devices have similar average performance across all locations. For example, devices 1 and 5 have relatively poor performance as compared to device 4, however, all three devices have consistent performance characteristics across all locations. We observe similar performance characteristics for applications and content providers in Figures 3.9(b) and (c) respectively. For example, applications 7 and 8 have worse performance across most locations. Similarly, content providers 9 and 16 have worse performance across all locations. Figure 3.9(d) shows heat map for combinations of devices and applications. Here, we observe weaker trends e.g., application 2 has worse performance across most devices, device 5 and 23 are weakly consistent across all applications. We observe no trends in Figure 3.9(e). Figure 3.9(f) shows that most combinations are invalid, represented as the darkest color with a value of zero on the color bar. This is because most applications are specific to some content provider except



(a) Locations - Devices (b) Locations - Applications (c) Locations - Content Pro-



Figure 2.5 Heatmap plots for combination of dimensions

browser applications. In summary, these observations show that locations have little impact on E2E performance and variations in performance occur across content providers, device types, and applications.

2.4 Modeling E2E Performance Dynamics

Our approach is to first build baseline E2E performance models and then detect and localize E2E performance anomalies based on these models. To build models, we first compute a two dimensional aggregate E2E performance matrix that captures the average performance along the dimensions of user locations, content providers, device types, and application types over each hour in a week. We choose a week as the duration because people have weekly diurnal patterns in their use of cellular services [125]. This aggregate E2E performance matrix is
used as an input predictor variable of a regression algorithm, which uses an iteratively reweighted least squares approach, to obtain a single baseline model for all users. Based on this single model, we filter out extreme outlier data points, which are mostly errors and noises introduced in our data collection process. A single model cannot accurately describe the performance of all E2E instances because some E2E instances have different performance than others. An *E2E instance* refers to the flows of a specific location, content provider, device type, and application type. To accurately model different groups of E2E instances, where within each group, all instances have relatively similar performance, but across different groups, instances have relatively different performance, we use an association rule mining approach to partition users into such groups. Furthermore, within each such group of E2E instances, there are subgroups where within each subgroup, all E2E instances have relatively similar performance, but across different subgroups, E2E instances have relatively different subgroups. Then, we build fine grained models for each group/subgroup and then detect and localize E2E performance anomalies based on these models.

Weighting Function	Talwar	Bisquare	Cauchy	Fair	Huber	Logistic	Welsch
Ordinary Least Square Method	20.06	20.06	20.06	20.06	20.06	20.06	20.06
Robust Regression Method	1.23	1.40	1.64	2.44	1.99	2.04	1.44

Table 2.1 RMSE values for different weighting functions and regression methods

2.4.1 Aggregate E2E Performance Matrix

Given TCP flow data for a certain time period of W weeks at a certain region as the training data, let L denote the total number of user locations, C denote the total number of content providers, D denote the total number of device types, and A denote the total number of application types. For each user location, we first calculate the average performance (packet loss rate or RTT) across all content provides, all device types, and all application types, for each hour in the W weeks of 24*7*W hours; and then, for each location and for each hour in a week of 24 * 7 hours, we calculate the median of the W values; thus, for each hour in a week of 24 * 7 hours, we obtain a vector of L median values. Similarly, for each content provider, we first calculate the average performance across all user locations, all device types, and all application types, for each hour in the W weeks of 24 * 7 * W hours; and then, for each content provider and for each hour in a week of 24 * 7 hours, we calculate the median of the W values; thus, for each hour in a week of 24 * 7 hours, we obtain a vector of C median values. We apply such calculation for device types and application types as well. In the end, we obtain a two-dimensional aggregate E2E performance matrix $E_A = [1..24 * 7, \{\mathbb{L}, \mathbb{P}, \mathbb{D}, \mathbb{A}\}]$, which has 24 * 7 rows and four columns with indices denoted $\mathbb{L}, \mathbb{P}, \mathbb{D}$, and \mathbb{A} . In this matrix, elements $E_A[i, \mathbb{L}], E_A[i, \mathbb{P}], E_A[i, \mathbb{D}], E_A[i, \mathbb{A}]$ are four vectors of L, P, D, and A median values, respectively, as we calculated above. This matrix E_A will be the input to the robust regression algorithm described below.

2.4.2 Coarse Grained E2E Modeling

Taking the aggregate E2E performance matrix as input, we build a single baseline model for all E2E instances, based on which we can remove extreme outlier data points. Note that these extreme outlier data points are not the E2E anomalies that we are looking for because such data points are mostly errors and noises introduced in our data collection process. We use robust regression for this baseline modeling because it can minimize the impact of extreme outlier data points on the produced model.

Robust regression uses the standard regression model $y = E_A \beta + \epsilon$, where y is the response vector of size 24 * 7, β is the coefficient vector of size 4, and ϵ is the residual vector of size 24 * 7. It computes a robust estimate of β such that the impact of extreme outlier data points on the produced model is minimized. To compute the coefficient β , we use the well known iteratively re-weighted least squares (IRLS) algorithm [69]. This algorithm first obtains the residual errors based on the standard least square method. Second, it uses a weighting function over the residuals to mark outliers. Third, it ignores outliers and reestimates regression coefficients. It repeats the above process until the difference of values of estimated coefficients obtained in two successive iterations approaches a minimum threshold. Mathematically, the robust estimates of β after n + 1 iterations can be represented as:

$$\beta^{n+1} = \arg\min\sum_{i=1}^{p} w_i^n |y_i - Q_{A_i}\beta|^2$$
(2.1)

Here w_i^n is the weight assigned to the i^{th} observation at the n^{th} iteration. The weight assignment is done through a weight function $w(r_i)$, where r_i is the scaled residual calculated using the method proposed in [46]. We choose the weighting functions that minimize the overall error by assigning less weights to outlying points. We used our training datasets to evaluate the performance of some well known weighting functions [69]. We calculate the Root Mean Squared Error (RMSE) of the regression models developed using seven different weighting functions. Table 2.1 gives a comparison of RMSE values obtained through Ordinary Least Square (OLS) method and the IRLS method. It is clear that the *Talwar* weighting function has the least RMSE value. This is because *Talwar* weighting function assigns a weight of 1 or 0 to each data instance. Equation 2.2 describes the weighting criteria, which is based on the scaled residual values r_i .

$$w(r_i) = \begin{cases} 1 & \text{if } abs(r_i) < 1\\ 0 & \text{otherwise} \end{cases}$$
(2.2)

Using *Talwar* in IRLS, a data instances with anomalous residual value will get a weight of 0, which means that all data instances that have zero weights will not affect the regression coefficients in the next iteration. Therefore, the baseline developed for the initial training set will not be affected by these outliers.

After we obtain the robust regression model, we calculate the residual error for all data points, plot their distribution, and then remove the data points whose residual error is outside of the 99.99 percentile of the residual error distribution. Figure 2.6 plots the cumulative distribution function (CDF) of the residual errors obtained from the original data, from which we observe that the noisy instances have very high percentage loss error reaching up to 500% with extremely low frequency. Figure 2.6 also plots the CDF of residual error obtained from the filtered data after we remove outlier data points. From the figure we observe that the maximum residual error that we obtain for the filtered data is less than 150. Whereas, we have several instances with very large residual error values.



Figure 2.6 Residual error distribution

2.4.3 Fine Grained E2E Modeling

We have defined aggregate E2E performance matrix, now we define E2E matrix. An *E2E* matrix denoted $E_I = [1..L, 1..P, 1..D, 1..A]$ is a four dimensional matrix where the four dimensions are L user locations, P content providers, D device types, and A application types. An element $E_I[l, p, d, a]$ in this matrix represents the E2E instances (or say flows) corresponding to user locations l, content provider c, device type d, and application type a.

Describing the performance of all E2E instances in the E2E matrix using only one model is not accurate enough because the performance of some E2E instances is significantly different from that of other E2E instances. For example, Figure 2.11(a) and (b) show the observed and predicted performance curves for two different cells of the individual E2E matrix E_I . We obtain the predicted performance curve using the single model in the above coarse grained E2E performance modeling. From Figure 2.11(a), we observe that for E2E matrix cell 1, the predicted performance curve are fairly consistent with the real one but from Figure 2.11(b), we observe that for E2E matrix cell 2, the predicted performance curve significantly deviate from the real one.

Our approach to addressing this issue is to partition E2E instances into groups such that



Figure 2.7 Real and predicted performance curves



Figure 2.8 No. of outliers

Figure 2.9 Single model

Figure 2.10 Multiple models

each group has distinct performance. The more groups that we partition, the more accurate models we can obtain, but at the same time, the number of models and the complexity will increase. On one extreme end, we have only one model for all E2E instances, which gives us the least accuracy and the least complexity; on the other extreme end, we have one model for each individual E2E matrix cell, which gives us the most accuracy and the most complexity. Our strategy to tradeoff between model accuracy and model complexity is to identify the E2E instance groups that have significantly different performance and then model these groups separately, leaving the rest E2E instances still being modeled by a single E2E model. Within each group, the performance of some E2E instances may be relatively different from that of others; thus, we apply this grouping strategy recursively among identified groups.

In this work, we identify E2E instances that deviate from the baseline model; then group such deviating E2E instances using association rule mining; within each group, we apply this strategy recursively. Next, we present the details of these three steps: deviating E2E instance identification, deviating E2E instance grouping, and recursive E2E instance grouping.

2.4.3.1 Deviating E2E Instance Identification

To identify the E2E instances that have deviating performance, we compare the real performance of each E2E instance with its predicted performance based on the baseline model for each hour in the 24 * 7 hours of a week. To take the standard performance deviation across W weeks into consideration, we compute a two dimensional matrix denoted \bar{E}_A that captures such deviation. Here \bar{E}_A differs from E_A only in that each value in \bar{E}_A is the standard deviation of the W values whereas each value in E_A is the median of the W values. In \bar{E}_A , the elements $\bar{E}_A[i, \mathbb{L}]$, $\bar{E}_A[i, \mathbb{P}]$, $\bar{E}_A[i, \mathbb{D}]$, $\bar{E}_A[i, \mathbb{A}]$ are four vectors of L, P, D, and Astandard deviation values, respectively. We feed \bar{E}_A to the same robust regression algorithm and obtain the standard deviation σ . For each E2E instance and each hour of the 24*7 hours of a week, if the real performance is too much out of the range of [predicted performance - σ , predicted performance + σ], we label this E2E instance as deviating for this hour. In this work, we use [predicted performance - $2 * \sigma$, predicted performance + $2 * \sigma$] as the guideline for quantifying "too much". For each E2E instance, if for a large percentage of the 24 * 7 hours of a week, this E2E instance has a deviating performance, we label this E2E instance as deviating. In this work, we use 50% as the guideline for quantifying "large percentage".

2.4.3.2 Deviating E2E Instance Grouping

Now each instance of the E2E matrix E_I has been labeled "not deviating" or "deviating". Visually, we mark non-deviating rules as white and deviating rules as black. The next step is to group such deviating E2E instances together based on their performance. In this work, we model the E2E matrix E_I as a transactional database and use the association rule mining technique to perform grouping. An instance $E_I[l, p, d, a]$ with color c is modeled as a transaction with five items $\langle l, p, d, a, c \rangle$. Given this transaction database as the input, we use the class based Apriori association rule mining algorithm to produce rules with only the color being the consequence. In association rule mining, a rule is of the form $\texttt{antecedent} \rightarrow \texttt{consequence}$ where antecedent and consequence are disjoint sets of some items. The meaning of the rule is that most people who buy antecedent also buy consequence. The quality of a rule is measured by two metrics: support and confidence. The support of a rule is the percentage of the transactions that contain all the items in both the antecedent and the consequence of the rule among all transactions in the database. The confidence of a rule is the percentage of the transactions that contain all the items in both the antecedent and the consequence of the rule among the transactions that contains all the items in antecedent and the consequence of the rule among the transactions that contains all the items in antecedent and the consequence of the rule among the transactions that contains all the items in antecedent and the consequence of the rule among the transactions that contains all the items in antecedent. We are only interested in rules with the color black as the consequence and therefore ignore all rules with color white. Each rule represents a group. For example, a rule

$\texttt{NewYork}, \texttt{Google}, \texttt{iPhone} \rightarrow \texttt{black}$

represent a group of all E2E instances whose user location is New York, content provider is Google, and device type is iPhone. This rule means that most E2E instances corresponding to user location New York, content provider Google, and device type iPhone, regardless of application types, have deviating performance from the baseline model. Here New York is just a metaphor for a specific RNC. For simplicity, we use

NewYork, Google, iPhone, *

to denote this group where wild card * denote all application types. The size of this group is A, which is the number of all application types. As another example, the size of the following group is D * A:

NewYork, Google, *, *

The size of each group is actually exactly the support of the corresponding rule. Note that some of the E2E instances in a group may be white. The percentage of the black E2E instances in a group is exactly the confidence of the corresponding rule. To tradeoff between model accuracy and model complexity, we want to find groups with a large support and a high confidence. In this work, we set the support threshold to be 0.1% and the confidence threshold to be 80%. We pay attention to only the rules whose support and confidence are above these thresholds and ignore the other rules. For simplicity, in the rest of this chapter, we use the terms "rule" and "group" interchangeably.

After we filter out rules who support or confidence is below the corresponding threshold, we still have too many groups to model each individually. We do not choose to reduce the number of rules by simply increasing support or confidence thresholds because a rule with larger support and/or confidence does not necessarily have a larger impact on the baseline model. Instead, we choose to select the groups that have the most impact on the baseline model. Next, we present an exclusion method and an inclusion method to quantify the impact of a group on the baseline model.

In the exclusion method, first, for the set of all E2E instances, denoted E_A , we use the simple regression algorithm to build the baseline model. Second, for each group of E2E instances R, we use the simple regression algorithm to build a model for $E_A - R$. Third, we calculate the absolute difference of the RMSEs of the two models for $E_A - R$ and E_A . The result is used to quantify the impact of group R. Note that here we use the simple regression algorithm, instead of the robust regression algorithm, because robust regression ignores outliers that have large effect on the RMSE.

In the inclusion method, let Σ denote the union of all groups; first, for $E_A - \Sigma$, we use the simple regression algorithm to build the baseline model. Second, for each group of E2E instances R, we use the simple regression algorithm to build a model for $(E_A - \Sigma) \cup R$. Third, we calculate the absolute difference of the RMSEs of the two models for $E_A - \Sigma$ and $(E_A - \Sigma) \cup R$. The result is used to quantify the impact of group R.

After we quantify the impact of each group on the baseline model, we can rank all groups based on their quantified impact. Note that groups may overlap. For example, the following two groups overlap:

For any two groups R_1 and R_2 that overlap, if R_1 is ranked higher than R_2 , then for all E2E instances in $R_1 \cap R_2$, we model them using the R_1 's model. Given the *n* ranked groups in the non-ascending order of their impact, denoted by R_1, R_2, \dots, R_n , for each $1 \le i \le n$, we build a separate model for group $R_i - \bigcup_{1 \le j \le i-1} R_j$. Note that for group R_i , if there exists $1 \le j \le i-1$ such that $R_i \subset R_j$, then $R_i - \bigcup_{1 \le j \le i-1} R_j = \emptyset$.

2.4.3.3 Recursive E2E Instance Grouping

Just like having one model for all E2E instances in E_I is not accurate enough, for some groups, having one separate model is also not accurate enough. Therefore, we apply our fine grained E2E modeling algorithm recursively on each group that we have identified as above. We now empirically show that through recursive E2E instance grouping, we achieve high modeling accuracy. We use a training data collected over a duration of six weeks (*i.e.*, W = 6). We calculate the aggregate E2E performance matrix using the six weeks data and build a single baseline model for all E2E instances. We then identify a first set of deviating E2E instance groups. For each of these E2E instance groups, we calculate the total number of E2E instances that are deviating from the single baseline model. In Figure 2.8, the black bars are the fraction of deviating E2E instances belonging to a particular group, and the grey bars represent the fraction of deviating E2E instances when a separate model is used for each of the E2E instance groups. For each of the E2E instance groups, we develop a separate set of fine grained models through recursion. The white bars are the fraction of deviating E2E instances in the original E2E instance group after separate sets of fine grained models are used for each E2E instance group after recursion. We observe that after at most three levels of recursion, we do not get any further E2E instance grouping. Figure 2.9 shows the CDF of residual errors obtained when a single baseline model is used. Figure 2.10 shows the CDF of residual errors obtained when a separate baseline models are used. We observe that recursive E2E instance grouping and fine grained modeling greatly reduce residual errors.

2.5 E2E Performance Degradation

2.5.1 Detection

So far we have discussed how to build E2E performance models based on training data. Now we present solutions to detect and localize degradations in an on-going data feed on an hourly basis. In order to determine whether a particular E2E instance in the on-going data feed has degrading performance in the latest hour, we compare the performance of that E2E instance in the latest hour with its predicted performance based on the fine-grained model for that E2E instance for the same hour. For a given E2E instance at a particular hour, if the value of its performance metric exceeds an upper bound then we label this E2E instance as "degrading". We use the predicted performance $+ 2 * \sigma$ as the upper bound for detecting performance degradation.

2.5.2 Localization

For each new hour in the on-going data feed, we first label each E2E instance as degradation or normal and then apply association rule mining algorithm on the E2E instances labeled as degradation. Unlike the training phase, we do not apply the association rule mining recursively. The output rules show the localization of performance degradations. For example, the following rule shows that for a particular hour, for all content providers, all device types, and all applications, the cellular users in the New York area are experiencing significant degrading performance for their cellular services.

$\texttt{NewYork} \to \texttt{degrading}$

This rule is useful for cellular network operators to investigate potential issues in their cellular services in the New York area. Again, here New York is just a metaphor for a specific RNC. For another example, the following rule shows that for a particular hour, for all locations, for all content providers, the cellular users of iPhone are experiencing significant degrading performance for their email application.

$iPhone, Email \rightarrow degrading$

This rule is useful for cellular network operators to contact iPhone manufactures to investigate potential issues in their email application.

2.6 Evaluation

In this section, we evaluate our approach using the operational data collected from a large US-based cellular service provider. The main challenge in our evaluation is lack of ground truth anomalies. Due to the prohibitively huge size of the data, it is practically infeasible to manually label all anomalous events. To address this challenge, we introduce some synthetic anomalies into the collected operational data to effectively evaluate the accuracy of our approach. Specifically, we first apply our approach on the operational data collected during six consecutive weeks to learn the fine grained E2E performance models and then use these models to detect and localize synthetic anomalies. Besides synthetic anomalies, we also present a few real-world anomalies detected and localized by using the learned fine grained E2E performance models. To identify real world anomalies we use past six weeks of data to build our training model and detect anomalies in the seventh week. Operational cellular networks consistently update their network by adding new cell towers, similarly new devices and applications are launched on daily basis. We update our E2E instance matrix E_I on weekly basis by adding new elements along the four dimensions of the matrix and reflect these updates in the aggregate E2E performance matrix E_A .

2.6.1 Synthetic Anomalies

To evaluate our system we introduce synthetic anomalies in the data. We created three sets of scenarios, in the first set we have 19 scenarios and in each scenario we introduce a one



Figure 2.11 Synthetic anomaly detection.

dimensional anomaly, the second set involves 26 scenarios with two dimensions and the third set involves 19 scenarios with three dimensions. In each scenario we introduced a one-hour anomaly that is associated with a group of instances in the huge E2E performance matrix. For example, a particular scenario introduces performance degradations at a particular hour, for all users accessing their email (application type) from Motorola V3 Razr (device type). Specifically, we first collect another week's operational data (different from the six weeks we used to learn the fine grained E2E performance models) and calculate the mean and standard deviation of the TCP loss ratio for each E2E instance using 168 hours of that week. Then for each E2E instance in each particular scenario, we create a one-hour synthetic anomaly by using $Anom = \mu + 2\sigma + |\mathcal{N}(0, 2)|$.

For each scenario, we first used the fine grained E2E performance models learned from six-week data to detect anomalies for each instance of E2E matrix and then use association rule mining to localize the root cause of all detected anomalies. As we use association rule mining for localization, we can tune the *support* and *confidence* threshold values to control the number of antecedents in the output rule. We choose support threshold values for each type of scenario separately. This is because on one hand the threshold may be too high for three dimensional scenarios which should be identified, but since they do not satisfy the high support threshold they are never selected as output rules. On the other hand, if the threshold value is too low then we may see a lot of undesirable rules. For example, for a minimum support threshold of 0, we obtain rules with the number of items in *antecedent* and *consequence* equal to number of items in a transaction, that effectively represents a single E2E instance. Therefore, we use three sets of support threshold values, one for each type of scenario. We use the frequency of occurrence of item sets in the transaction data set to select the support thresholds. We choose the frequency of least frequent 1-itemset in the data as the minimum support threshold for 1-dimensional scenarios. Similarly, we choose the frequency of least frequent 2 and 3-itemsets as the minimum support threshold for 2 and 3-dimensional scenarios respectively.

We use confidence as our evaluation metric for the rules obtained through association rule mining and select the top ranked rule obtained from association rule mining according to their confidence values. As we can see from Figure 2.11(a), Figure 2.11(b) and Figure 2.11(c), all 64 synthetic anomalies that we injected into the collected operational data are successfully detected and localized. We get all scenarios as association rules with 90% confidence for one dimensional scenarios. Figure 2.11(a) shows the confidence of top 50 rules obtained for one dimensional scenarios using a minimum confidence threshold of 10%. The red bars represent the synthetic scenarios introduced in the data. The blue bars represent the false positives which are identified due to noisy instances in the data. From this figure we observe that by selecting a minimum confidence threshold of 40% we do not obtain any false positives. Figure 2.11(b) and 2.11(c) shows the confidence of top 50 rules obtained for two dimensional and three dimensional scenarios respectively, using a minimum confidence threshold of 0.1. We select confidence threshold using the optimal operating point of the Receiver Operating Characteristic (ROC) curve. For all three scenarios we obtain ROC operating points separately and choose confidence threshold that gives minimum false positive rate and maximum true positive rate. We achieve 0 false positives for a minimum confidence threshold of 70% and 80% for two and three dimensional scenarios respectively.

2.6.2 Anomaly Detection in the Wild

In this section, we characterize the performance anomalies detected on an operational network over a period of one week in August 2014. Through this characterization analysis, we will explain that how frequent E2E performance anomalies are and how often multiple dimensions are the root cause of performance degradation. We implemented and deployed our system on an operational network and detected performance anomalies for packet loss ratio and RTT separately. Overall, we monitored 78 different regions, 51 device types, 13 application types and 36 content providers.

It is interesting to see that how often a single dimension is responsible for performance degradation. We use the output rules from the localization step to count the number of single dimensional and multi-dimensional anomalies. For each rule, we count the total number of all E2E instances and the number of E2E instances marked as degrading, which belong to that rule. Table 2.2 gives the number of one dimensional anomalies detected on hourly basis. For each dimension the number of anomalies is the number of E2E instances which were marked as degrading in the detection and localization process as discussed in Section 2.5. We also report the average confidence that is the fraction of all E2E instances which were marked as degrading. We can see that there are no performance anomalies in the user location dimension, which means that a single user location cannot be blamed for the anomalies occurring during the one week's time period. We observe that one dimensional anomalies are rare for content providers and device types. This is mainly because content providers use content distribution networks for serving content to remotely located users. It is very unlikely that the performance of a content provider will degrade across all user locations. Therefore, a problematic content server can only affect the performance of a subset of user locations. For applications, single dimension anomalies are relatively frequent. This observation highlights the performance impact of application patches and bug fixes that are released periodically and frequently and bugs in these updates can cause performance issues at all user locations, across all device types and content providers.

	Dim	Loc	Dev	App	CP
Loss	Anom	0	5	98	27
	Conf	0	0.80	0.68	0.67
RTT	Anom	0	11	82	0
	Conf	0	0.82	0.68	0

Table 2.2 Single dimension anomalies

	Dim	L,D	L,A	$^{\rm L,P}$	D,A	D,P	A,P	L,D,A	$_{\rm L,D,P}$	$_{\rm L,A,P}$	D,A,P
Loss	Anom	134	0	215	79	263	0	378	44	233	962
	Conf	0.94	0	0.93	0.76	0.83	0	0.82	0.80	0.76	0.78
RTT	Anom	57	13	56	280	355	0	32	20	444	4627
	Conf	0.80	0.85	0.84	0.71	0.83	0	0.82	0.96	0.72	0.78

Table 2.3 Multi-dimension anomalies

Next we look at multi-dimensional anomalies reported in Table 2.3. We observe that anomalies involving multiple dimensions are more frequent as compared to single dimension anomalies. We detected a total of 8415 E2E instances as anomalous. Our analysis reveals that in 83% of these E2E instances content provider was one of the dimensions, 86% involved device types and 86% involved application types. Overall, in 80% of these instances, the RNC or location dimension was not involved. In other words, 80% of the time the anomalies are not because of problems at the cellular network. This composition of E2E performance anomalies shows that most of the time content providers, device types and applications are involved in the performance degradations. These results are extremely useful for network operators as they highlight the nature of majority of performance anomalies. Additionally, the findings point network operators to the problematic dimensions and reduce the search space of root causes. Verification of detected E2E performance anomalies requires manual analysis of hundreds of time series. The manual analysis acts as the ground truth of the detected E2E instance. For each E2E instance, we plot all the time series and verify the occurrence of anomalies in those time series. We also verified the network anomalies in the cellular network outage reporting system. However, due to large manual effort required to verify each anomaly from the network outage reporting system, we verify two real anomalies.



Figure 2.12 Real-world anomaly 1: root cause outside the cellular network



Figure 2.13 Real-world anomaly 2: root cause inside the cellular network

2.6.3 Real-World Anomalies

In the section, we present two interesting real-word anomalies detected and localized by the learned fine grained E2E performance models (same as the synthetic anomalies) and the possible underlying root causes.

The first real-world anomaly involves three dimension namely, content provider, device type and application type. Our system detected and localized a one-hour anomaly across all 78 user locations (corresponding to 78 RNCs) but specific to apple.com (content provider), iPhone 4s (device type) and browsing (application type). Figure 2.12 shows the time series plot of the TCP loss ratio for all 78 user locations , apple.com, iPhone 4s and browsing. We also plot the predicted performance and the upper bound of the TCP loss ratio for apple.com, iPhone 4s and browsing across all locations. One can clearly see that the anomaly occurred around the 8th hour in the week for all 78 user locations. The figure also shows the confidence values that we obtain for this particular case. We observe that the confidence value peaks during the 8th hour, indicating that all E2E instances belonging to apple.com, iPhone 4s and browsing were marked as anomalous. As the peak in loss ratio is across all locations and specific to apple.com, iPhone 4s and browsing, the root cause is most likely to be outside of the cellular network and might be apple related. There are other time instances where we observe non zero confidence values, indicating anomalies occurring across a small fraction of E2E instances belonging to apple.com, iPhone 4s and browsing.

In the second real-world anomaly, only one dimension is involved. Our system detected and localized a 15-hours anomaly for four user locations across all content providers, device types and application types. Figure 2.13 shows the time series plot of the TCP loss ratio for all locations aggregated on other 3 dimensions. We only picked one user location to plot the predicted performance and the upper bound of the TCP loss ratio for that user location, apple.com, iPhone 4s and browsing due to space limits. One can clearly see only 4 curves (corresponding to 4 RNCs) had a significant spike starting around hour 108. Besides the 4 significant spikes, there are several locations whose actual performance is exceeding the upper bound. The figure also shows the confidence values that we obtain for the performance degradation across different locations. We observe that the confidence value peaks to 0.7, indicating that 70% of the locations were marked as anomalous. As a contrast to the first real-world anomaly, the anomalies here are only specific to only user locations regardless other 3 dimensions. Thus the root cause here is most likely to be something inside the cellular network impacting the user locations (corresponding to RNCs).

2.7 Conclusions

We make the following key contributions in this chapter. First, we design and implement a comprehensive and holistic measurement system that monitors E2E service performance across four dimensions - user locations, content providers, device types, and application types. Second, we propose fine grained models that capture the normal performance for every combination of user locations, content providers, device types, and application types, and use the models to detect performance degradation. Third, we propose an association rule mining based approach to localize performance degradation. Fourth, we use both real network traces and synthetic performance degradation to show that our method is highly effective.

3 Assessing Impact of Cell Tower Outages in Cellular Networks

3.1 Introduction

3.1.1 Background and Motivation

The ubiquity of cellular data networks has fueled the rapid increase in the number of cellular network users that connect to the Internet for an increasingly diverse set of applications ranging from entertainment, such as streaming videos and online gaming, to commercial applications, such as online shopping, to even some more critical applications, such as VoIP calls. Cellular network operators need to maintain high reliability and performance.

Despite the constant efforts by cellular data network providers in hardening and upgrading their network infrastructures, it remains a hard challenge for cellular network operators to identify service disruptions and assess impact of service disruptions from end-users' point of view. Under normal conditions, each user is typically "connected" to a set of nearby cell sites via multiple radio links for fast data transmission and service redundancy. The users are connected to multiple sites to facilitate seamless handoffs. With such an inherent service redundancy mechanism built in, the impact of a site outage may be "absorbed" by the surrounding sites through a rippling series of signaling, with the result being that users perceive no service impact. On the other hand, it is also possible that users associated with fully functional sites may suddenly suffer performance degradations resulting from the sudden migration of traffic load from a nearby site that failed. This makes the impact analysis of a site failure dependent on its nearby site locations, antenna configurations, power settings, user mobility, local landscape features (e.g., hills, buildings), and a wealth of other potential factors. Detailed modeling of each of these features can be extremely challenging, if not impossible.

3.1.2 Limitations of Prior Art

The state-of-art monitoring systems deployed in cellular networks mostly rely on cell site level measurements [117, 94]. As a result, granularity of current monitoring systems is limited to the level of cell sites, which is typically an aggregation of tens or hundreds of users depending on the locations of the cell sites. Unfortunately the cell site level measurements are insufficient for cellular network operators to truly understand the fine-grained impact of cell site outages from users' perspective due to the unique challenge in cellular networks stemming from the built-in redundancy of radio links. Typically an user is within the radio reach of multiple cellular sites and upon the failure of his or her associated site, he or she may switch to get service from other nearby cell sites [67]. This type of radio migration during outage is very hard to be estimated without user level measurements. Due to scalability constraints, only spatially and temporally sampled user level measurements are collected, which are typically insufficient for cellular network operators to determine what user perceived network quality in the field given one or multiple cell site outage [123]. Even worse, the accurate locations are hard to obtain for the sampled user level measurements as cellular users are moving around most of the time [10, 7]. Consequently, the impact analysis result may be quite inaccurate, for example when the operator misjudges the failure scenario, or under- or over-estimates the user impact. When inaccurate impact analysis result is used in outage prioritization and escalation, many severe outages may not receive adequate attention from various teams of operators, vendor supports, customer care and public relation officials, and executive personnel, causing prolonged user dissatisfaction.

3.1.3 Proposed Approach

In this work, we propose to leverage the existing M2M devices deployed in the field as the approximation of human users to enhance service quality monitoring in operational cellular network. We refer to this approach as *M2MSourcing* in this chapter. Our key idea is to monitor communication activity of M2M devices over the cellular network infrastructure and identify gaps in the activity to estimate quality of service perceived by human users.

The ubiquity of cellular data networks has attracted vendors from the (M2M) device market to develop innovative applications that use operational cellular networks. Popular applications of M2M devices in cellular networks include: home security provided by DIGI-LIFE, utility monitoring through Smart Grid and transportation services by OnStar. Some of these M2M devices such as those used by home security applications, do not *sleep* and perform automated data transfer to an application server in the Internet. This automated data transfer between machines is commonly known as Machine Type Communication (MTC). Service disruptions in cellular networks directly impact MTC activity of M2M devices. Additionally, some applications of M2M devices require the M2M device to be fixed at a particular location. Examples of such applications are home security, fire alarm, smart meters etc. These applications require the device to be installed at a fixed location and then communicate periodically to a central server through cellular data network. These devices serve as ideal candidates for monitoring network service quality at specific geographical locations. Given the location of stationary M2M devices, network operators can monitor MTC activity of these devices to estimate service quality around those coordinates. Therefore, we propose to utilize existing M2M devices deployed in the field as sensor nodes for monitoring network service quality and analyze user impact at specific geographical locations.

In this chapter we design and evaluate a M2MSourcing based cellular network monitoring system called M2MScan that monitors communication activity of M2M devices to analyze and estimate the user impact during cell site outages. For monitoring purposes, we want to identify a significant number of M2M devices satisfying certain criteria such as stationarity, communication frequency (always awake) and spatial distribution. For estimation of user impact, we want to utilize the communication pattern of identified set of stationary M2M devices for performing measurements of network quality in different coverage regions. Our approach consists of three steps: identification of candidate M2M devices, detection of network outages, and measurement of user impact. First, we use call-data records of a major cellular service provider to identify a set of M2M devices where each device in the set should satisfy the following two criteria: (i) the M2M device should be fixed at a particular location and should be stationary (ii) the M2M devices should communicate continuously with their application server. In this chapter we refer to this set of M2M devices as stationary and active M2M devices or simply qualified M2M devices. We use the qualified M2M devices as network sensors to monitor network quality. Second, to detect network outages, we develop a spatial map of candidate M2M devices that allow us to monitor network quality at multiple locations within a cell sector. We monitor the communication activity of all M2M devices in a given region to detect outages. Third, we use a non-parametric statistical approach for identifying those devices in the region which have discontinuities in their communication pattern due to an outage. Additionally, our approach allows us to measure the duration during which a device was offline. Using this information we measure the spatial and temporal nature of impact of a service disruption on users. Our operational experience reveals that our M2MSourcing based system M2MScan is able to provide cellular network operators with fine-grained spatial and temporal aspects of users impact for cell site outages.

3.1.4 Technical Challenges and Solutions

We face several key technical challenges in monitoring M2M devices for assessing service quality. The first challenge is to accurately infer location of candidate M2M devices. Location of candidate M2M devices is required for creating a spatial map. To deal with this challenge we use approximated location samples which are obtained by using signal strength of a device. The signal strength is measured at multiple site locations and then through triangulation the location is approximated. We use multiple location samples taken over a period of one month for each device and then select the most representative location sample by identifying the medoid of the measurements.

The second challenge is to infer the customer impact of an outage which is measured in terms of outage start time and duration. Typically, the documented outage start times reflect the times when outages are first reported by users or they are reported through existing outage reporting systems. Users calling times are inherently inaccurate because users may experience bad service at a time other than the actual start time and users' calls are strongly contextual to users' tolerance levels. Similarly, existing reporting systems have to go through a series of manual verifications to finally report the outage start times. To deal with this challenge we utilize the communication activity of candidate M2M devices to infer the outage start time. We denote the set of devices in a region R as D_R . For each device $d \in D_R$, we use a p-value based statistical approach to compute a score table $P_{i\times W}^d$, where each row of the table P is identified by a start-time value and each column is identified by a duration value. Here each entry of the table $P_{i\times W}^d$ represents p-value at a given start time and duration. We compute the number of impacted devices for different values of $\langle i, W \rangle$ pairs. The pair with maximum number of impacted devices represent the outage start-time and duration.

The third challenge is to quantitatively evaluate the effectiveness of our approach. To deal with this challenge, we introduce synthetic cell site outages in real data and used M2MScan to identify these outages and estimate their user impact. We evaluated M2MScan using three different metrics namely; accuracy of estimating outage starting time, outage duration, and user impact. We also use real network outage reports collected over a period of 6 months for evaluation purposes. We compare the reported outage hours with those identified by M2MScan and then measure the user impact using frequency histograms.

3.1.5 Key Novelty and Advantages

To the best of our knowledge, monitoring user experience via M2M devices in a large scale operational cellular network has not been previously studied. M2MSourcing is a simple cost-effective yet pragmatic approach to measure the user impact of cell site outages by discovering abnormal communication activities of stationary M2M devices. This approach has several advantages over state of art network monitoring tools. First, there are no changes required in the existing infrastructure. Prior approaches like Jigsaw and WiScape deploy sensor nodes in the field to measure network performance [42, 123]. This kind of deployment for measuring network performance for nation wide operational cellular networks is infeasible. Second, a large percentage of M2M devices are deployed and maintained by third parties, therefore from the perspective of cellular networks there are no maintenance costs associated with these M2M devices. Thus M2M devices deployed in the field are free user experience sensors which can be used to closely measure the user impact of cell site outages. Third, we use stationary M2M devices for monitoring network conditions. Prior work has shown that mobility of end-user equipment can cause up to 79% difference in measured throughput [55, 106]. Utilizing stationary M2M devices allow us to obtain consistent measurements over time. Fourth, we filter out M2M devices which are active for an indefinite period of time. This has advantage over prior crowd-sourcing based approaches, as M2MSourcing is not dependent on user input to obtain measurements.

3.2 Related Work

Prior work on measuring service quality of large operational networks has adopted two popular measurement methodologies; user feedback, and user-level performance measurements. Authors in [79] use a tool named HostView to obtain user feedback and then predict service quality through passive measurements of system and network performance metrics. In [78], Jin *et al.* study the effectiveness of using a smartphone applications for reporting service problems in large operational networks. The authors use a Location based reporting tool (LRT) to obtain user feedback about service quality issues in cellular networks. User feedback based methodologies are limited by the accuracy and completeness of information provided by users. Furthermore, in such cases a feedback is recorded only when the user initiates the feedback process, thus limiting the visibility into network conditions over time. Our work utilizes M2M devices which communicate continuously, thereby allowing us to analyze network service quality at any given time.

There is a significant amount of work on assessing network quality by measuring different network performance indicators pro-actively [55, 106, 100, 127]. Q-score is another service quality assessment framework that quantifies service quality by correlating network performance indicators with user feedback in the form of customer troubled tickets [128]. Netradar is a mobile network measurement platform that performs active and passive user level measurements to obtain insights about user-perceived service quality [129]. It measures signal strength and relate it to user service quality. Jigsaw and WiScape are client assisted wide area wireless network performance monitoring frameworks that deploy monitoring nodes in the field to measure network performance [42, 123, 33]. In this work, we propose to utilize existing M2M devices already connected to the Internet through an operational cellular network for monitoring user experience. This allows us to monitor network health at any given time at any given location. To the best of our knowledge M2MScan is the first tool to utilize M2M devices for network monitoring.

3.3 Background and Data Collection

Figure 2.3 shows the UMTS cellular network architecture. A UMTS cellular network consists of two main sub-networks; the radio access network and the core network. Radio access network consists of NodeBs which provide radio access link to the user equipment. The UMTS core network consists of two main components the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN). GPRS Tunneling Protocol (GTP) is used at the Gn interface to send and receive data to external networks. The GTP protocol is an IP protocol used in UMTS and LTE core networks. In a UMTS network, a Packet Data Protocol (PDP) Context is established between SGSN and GGSN for packet transfer to and from the User Equipment (UE) [15]. At the UE side, connectivity to the Internet is achieved by establishing a radio resource channel (RRC) with a nearby cell site. Typically, due to built-in redundancy of cellular network architecture, multiple cell sites are available for RRC establishment.

Despite the built-in redundancy of cellular networks, cellular network users can experience poor or no service due to a variety of reasons. One of the major reasons is cell site outages which may occur due to bad weather conditions, power outages, cable cuts etc. During a cell site outage, users in the coverage regions would connect to nearby correctly functioning towers. Thus a cell site outage may not have any measurable user impact i.e., users do not experience service disruption. In worst cases multiple sites located in the same geographical region may fail together and render services inaccessible to some users.

3.3.1 Dataset

In this study we use data collected at the core of a nation-wide cellular network provider in the United States. The data consists of: 1- M2M device activity over time 2- Approximate M2M device location 3- Cell site outage information 4- Blackout regions where users experience no service during cell site outages. Note that all data sources were anonymized to protect the privacy of cellular network users.

Device activity. We use call and data records maintained by the cellular network for analyzing device activity. For each device the data set consists of information about data and call sessions. For data records, the measurement system is deployed at the core of a cellular network, it monitors the PDP context session of each device. A PDP context that is established by a device can remain active for hours depending on the activity of that device. The measurement system creates a record whenever a connection is torn down or the session length reaches a specified time limit, typically 1 hour. For data sessions, each record consists of time stamp in GMT at which the record is created, anonymized device identifier, PDP context duration in seconds, bytes uploaded and downloaded, connection type, cell and cell-sector identifiers. We use this information to monitor the communication activity of devices connected to the cellular network. Communication activity is a good indicator for differentiating a smartphone from a M2M device. Several M2M applications required M2M devices to communicate continuously to the application server. As a result PDP contexts established by these devices remain active 24 hours a day and 7 days a week. Therefore, by monitoring communication activity of M2M devices for a reasonable period of time we can identify *active* M2M devices. Furthermore, we use activity information for detecting cell site outages. When an outage occurs, M2M devices in the outage region are unable to access the cellular network as a result the measurement system does not record any activity for the impacted M2M devices. By looking at the time series, we can identify gaps in the communication activity of groups of M2M devices. We also measure impact of an outage by computing the fraction of M2M devices impacted out of all the M2M devices in the outage region. We use this fraction as an estimate of user impact.

Device location. For each device this data set consists of approximate latitude and longitude information. The approximate location coordinates are computed by measuring the signal strength of the device at various cell sites and then approximating the location through triangulation [29]. Each record consists of the time in GMT at which the measurement was taken and the approximate latitude and longitude information. The measurement system records multiple location samples of devices over time. Multiple location samples taken over different time instances are useful for identifying if a device is moving or stationary. Using these location samples we can create a map of active and stationary M2M devices which can act as sensors for monitoring network health.

Network coverage reports. We also have information about network coverage of cell sites. Given a set of sites we can obtain geographical regions which are expected to be



Figure 3.1 Spatio-temporal characterization of M2M devices

completely out of service (blackout regions). Additionally, we use network outage reports recorded by various network performance monitoring tools and by customer calls. Each record consists of network outage ID, outage start time, outage end time, type of outage(cell site, RNC, etc), list of cell sites affected, technology of the cell sites i.e. 2.5G, 3G, 4G or LTE, and cause of the outage (power outage, weather, etc). Note that not all types of outages affect the device connectivity, therefore we only focus on those types of outages which directly impact user activity. We use this information to evaluate M2MScan.

3.4 Spatio-Temporal Characterization

In this section we characterize devices connected to a cellular network according to their communication activity and mobility patterns. Our characterization analysis allow us to differentiate M2M devices from smartphones. We are interested in a specific set of M2M devices which satisfy the following two requirements. First, the devices should be stationary. M2M devices used for metering, building security, fire alarm are installed at fixed locations and act as reliable information sources. Typically, these devices use only cellular data services and are inactive over the voice plane. A major benefit of monitoring stationary devices is that service quality landscape can be characterized more accurately. Second, the devices should be active 24/7. M2M devices communicating continuously provide useful hourly information about the network accessibility conditions. Discontinuities in the communication activity are clear indications of potential network accessibility issues such as cell site outages. In the rest of this chapter we use the term *qualified* for devices which satisfy these requirements.

Mobility characteristics. We first identify devices which are inactive over the voice plane of cellular service for a period of one month. This gives us a set of devices which are active only over the data plane. Next, we utilize the location samples collected during a one month period and calculate pairwise distances between all location samples of each device. For each device we obtain the median value of all pairwise distance values. Figure 3.1(a) show the cumulative distribution function of the median distance computed for each device. From this figure, we observe that 40% of the devices which are inactive over the voice plane have a distance of less than 100 meters. We also observe that a small percentage of the devices (< 5%) have a distance between 100 to 1000 meters. Therefore, we consider all those devices stationary which have the median pairwise distance less than a threshold of 100 meters. We also look at the density of stationary devices. Figure 3.1(b) shows the CDF of number of devices present in a single hexagonal coverage region (cell sites). The figure shows that 80% of cell sites have at least 10 stationary device.

Communication activity. Besides stationarity we require that the qualified set of devices should also communicate continuously overtime. This is necessary for continuous network quality monitoring. A discontinuity in the communication pattern of a device indicates that the device is offline. The discontinuity can be either due to device application behavior or



Figure 3.2 Active device density

poor network service. A device which has discontinuities in its communication under normal conditions is not a reliable candidate. For monitoring network service quality in the face of a cell site outage, we want to identify devices which went offline due to the outage and not because of the application behavior. Therefore, we filter out devices which have discontinuities in their communication pattern. To filter out less active devices we simply look at the number of records generated by each device per hour. Figure3.1(c) shows the CDF plot of the number of records per device over a period of one week. The figure shows that approximately 18% of the qualified devices generate at least one record in every hour i.e., 7% of the original set of devices used in Figure 3.1(a). These devices serve as our qualified set of devices which are stationary and active over the data plane.

Geo-spatial characteristics. We next step look at the spatial distribution of the qualified devices. Figure 3.1(d) shows the CDF plot of the number of qualified devices present in a cell. The figure shows that more than 60% of cell sites have at least one device which is stationary and communicates continuously. In Figure 3.2 we visualize the county-wise geographical map of device density. The colored regions do not have M2M devices that satisfy our criteria. On average majority of the regions have 10 to 50 devices per cell sector. Counties with large metropolitan regions have device densities greater than 100 devices per cell sector. These results show that sites located in large metropolitan regions have higher device density that allow service quality monitoring at higher granularity.



Figure 3.3 Overview of system architecture

3.5 System Design

Our approach is to monitor communication activity of stationary M2M devices to measure service quality and detect cell site outages. This approach has several advantages over traditional monitoring approaches. First network operators can obtain a service quality landscape at a higher resolution. Second, activity patterns of M2M devices can be used to detect cell site outages. Third, M2MSourcing allows network operators to measure customer impact of an outage. Figure 3.3 shows the overview of M2MScan system architecture.

3.5.1 Activity Profiling

For each device we measure its activity on hourly basis by looking at number of bytes uploaded and downloaded in that hour. At any given hour, if the total number of bytes uploaded or downloaded is zero then we consider the device as in-active. We consider any in-activity as an impact on user experience. We can infer the impact duration on hourly basis by looking at the in-activity start time and end time. Figure 3.4 (a) shows time series plot of some example devices impacted due to a cell site outage. The discontinuities in the time series plot of each device represent the durations during which devices were unable to download/upload data to their application server. We make two main observations: First, the impact duration of individual devices located within the same outage region can vary. From the figure we observe that all three devices were impacted for different time durations.



Figure 3.4 Example devices

Second, a device may be able to communicate inconsistently during the complete outage duration. From the figure, we observe that Device-A is impacted during two different time windows. These observations highlight the dynamic nature of impact of network outages on cellular network users. M2MScan allows network operators to monitor variation in impact durations and identify user impact at a higher granularity both in space and time.

3.5.2 Location Profiling

We use location samples collected for each device and obtain the most representative location sample as a device's location. For each device all location samples form a cluster on the geographical map. We use medoid to identify the most representative sample of each cluster. A medoid is defined as the most centrally located point of a cluster. Figure 3.4 (b) shows an example of location samples and the medoid of a device. The figure also shows the sites near the measured samples, where majority of samples of each device are clustered together.

3.5.3 Quantifying Temporal Impact

In this section we discuss our approach for inferring the temporal impact of network cell site outages. We denote the set of devices in a region R as D_R . For each device $d \in D_R$, we monitor the device activity in terms of traffic volume on hourly basis. Assuming two time instances a and b, where b - a = N hours, we compute sum of traffic volume using a sliding window of W hours. This gives N - W + 1 samples of sum of traffic volume between a and b. At time i we denote the sample as $S_{i,W}$, where i represents the starting time index of sample window and W represents the window size. Next we rank the N - W + 1 samples while breaking ties. We use the N - W + 1 ranks to compute p-value at various time instances between a and b using Equation 3.1.

$$pval(i, W) = \frac{rank(S_{i,W})}{N - W + 1}$$
(3.1)

Here $i \leq b-W$, and $W \ll N$. For each device we compute p-values for N values of i and Mvalues of W. Let $P_{i,W}^d$ contain p-values for all the set of pairs of i and W for device d. We use all sets of pairs $P_{i,W}^d \notin D_R$ to compute a frequency histogram of devices denoted as $H_{i,W}$. We select pairs of values of i and W for which we get p-values less than a specific threshold th. We select the threshold th according to the total number of samples between the two time instances a and b i.e., W/(N - W + 1). The threshold is such that the samples with smallest rank values are selected. The pair with maximum frequency represent the outage start-time and duration $\langle S_t, \mathbb{D} \rangle$, during which maximum number of devices were affected. Note that the sets of devices belonging to different pairs of start times and durations may not be disjoint i.e., a device may contribute to the frequency of different start time and duration pair. We consider devices belonging to the pair with highest frequencies for quantifying the user impact in space. Algorithm 1 describes our methodology for obtaining estimates of outage start-times, duration and user impact.

3.5.4 Quantifying Spatial Impact

To quantify the impact of a network outage we utilize the list of devices obtained through the p-value computation. First we obtain a spatial map of impacted devices, then we use a 2D Gaussian kernel to obtain a quantifiable measure of impact regions. For a device $d \in D_R$ we compute the impact score in its surrounding region using 2D Gaussian kernel centered over

Algorithm 1 Estimating outage start time and duration

Result: $\langle S_t, \mathbb{D} \rangle, |D'| = \texttt{Estimate}(D_R, th, M)$

Input: (1) Set of devices D_R in region R

- (2) P-value threshold th
- (3) Maximum duration M

Output: (1) Start time, duration pair: $\langle S_t, \mathbb{D} \rangle$,

(2) No. of devices impacted |D'|

the device location. We obtain the impact score as the value of function $G_d(x, y)$ defined as follows:

$$G_d(x,y) = \frac{1}{2\pi\sigma_d^2} e^{-\frac{x^2 + y^2}{2\sigma_d^2}}$$
(3.2)

Here x and y represent the coordinates at which the score is being computed and $x^2 + y^2$ is the squared distance between the device and the location. We define σ_d as the distance between the device and its nearest neighbor. The impact score of a device at x = 0 and y = 0, i.e., at the device's location is simply $\frac{1}{2\pi\sigma^2}$. To compute the impact score at any given location x' and y', we compute the scores $G_d(x, y) \forall d \in D_R$. Using the values of $G_d(x, y)$ we quantify the strength of outage impact at any particular location.



Figure 3.5 Estimation accuracy of start times and duration of synthetic outages

3.6 Evaluation

We use real data collected at a nation wide cellular network operator in our evaluation. First, we introduce synthetic network outages in the real data and evaluate our system using three different metrics: network outage start time, network outage duration and outage impact in terms of fraction of devices affected by the outage. Second, we deployed our system in an operational network and detected/validated network outage events and their impact. For validation purposes we uses external sources of information such as tickets generated by user complaints and network outage databases maintained by the cellular network operator.

3.6.1 Synthetic Network Outages

A naïve way to simulate an outage in a geographical region is to remove activity records of all qualified devices in that region for a specific duration. In real outage scenarios, devices in an outage region typically connect to nearby working sites and remain active. Therefore, all devices in an outage region are not impacted. To allow for real outage simulations we use network coverage reports to obtain blackout regions. These are geographical regions where customers experience complete network outage when cell sites in those regions fail.

Evaluation data: We identified six geographical regions covered by 102 sites and 1881 qualified devices. Each geographical region belongs to a large metropolitan city in the United States. The smallest region consists of 3 sites, and the largest region is covered by 37 sites. We obtain blackout regions for each set of sites. Approximately, 5% of the qualified devices are in the blackout regions. Using these sets of sites and devices we created 120 different outage scenarios. For each region we used device activity data for a period of one week. The length of a real network outage typically ranges from a few minutes to tens of hours. Since, our dataset consists of hourly measurements we introduce synthetic network outages of time duration ranging from 1 to 15 hours.

Outage scenarios. In each outage scenario, we choose a set of sites and introduce synthetic network outages at a given starting time and duration. For a given set of sites Tin a geographical region we obtain a set of blackout regions R from the network coverage report dataset. These regions are 100×100 meter boxes where users are likely to experience absolute service inaccessibility if the cell sites in T are out of service. Next, we obtain a set of qualified devices D, served by the sites in T and identify a subset of devices $D_b \in D$ which are located in the blackout regions. Finally, we introduce network outage of duration d, at time t by removing activity records of devices in the set D_b . For simulation purposes we assume that devices not in the blackout region are able to switch to a nearby site not in the set T and continue communicating over the data plane seamlessly. For each set of sites T, we created 20 different scenarios with four different durations and five different starting


Figure 3.6 Estimation accuracy of user impact in synthetic outages

times. Each outage scenario serves as the ground truth for evaluation purposes. Finally, we evaluate detection accuracy by recording the time and duration parameters returned by M2MScan for each outage scenario.

Evaluation metrics. For a given outage scenario M2MScan returns the $\langle S_t, \mathbb{D} \rangle$ pair that has maximum number of impacted devices. We compare the $\langle S_t, \mathbb{D} \rangle$ values for each scenario with the ground truth and compute the estimation accuracy. An accurate estimate of outage start time duration is essential for measuring temporal impact of an outage. Here we define estimation accuracy as the percentage of outage scenarios whose $\langle S_t, \mathbb{D} \rangle$ parameters were correctly identified.

We also evaluate the accuracy of estimating user impact. We measure user impact as the percentage of devices affected during the outage time. For evaluating accuracy of user impact estimation, we compute the fraction of impacted devices identified by M2MScan out of the total number of impacted devices. We evaluate user impact by looking at the fraction of impacted devices correctly identified by M2MScan out of the total number of devices actually impacted. Figure 3.6 shows the accuracy of estimating user impact. We observe that accuracy of user impact increases with p-value threshold because higher p-values allow inconsistent devices to be classified as impacted.



Figure 3.7 Average of absolute difference in timings



Figure 3.8 Device impact of real outages

Simulation results. Figure 3.5 shows the accuracy of estimating outage starting time and duration. We report estimation accuracies for four different durations. Each group of bars represent accuracies for different p-value thresholds. We also draw error bars which represent the percentage of scenarios whose estimations were incorrect. Top portion of an error bar represents percentage of scenarios whose parameter values are more than the actual value. Similarly, bottom portion of and error bar represents percentage of scenarios whose parameter values are less than the actual value. For example, Figure 3.5 (a) shows that the accuracy for outage scenarios with duration equal to one hour is approximately 67%.



Figure 3.9 Heatmap visualization of real outage for different p-value threshold



Figure 3.10 A special case of real world site outage.

3.6.2 Real Network Outages

We implemented and deployed M2MScan at the core of a cellular network and analyzed outages during a two month time period. We analyzed 14 different outages of varying time duration and different geographical locations. In each outage more than 3 cell sites are involved. For all outages, the average device density is 38 devices per cell site with a maximum of 145 qualified devices per cell site and minimum of 20 device per cell site.

We utilize information available in network outage reports as ground truth of real outages. We evaluate the accuracy of M2MScan on real network outages for different p-values. Figure 3.7 shows average of absolute difference between estimated and actual start time and duration values. For large p-values we observe large average difference that indicates lower estimation accuracy. These results correlate with the results obtain through simulations of synthetic outages. For each network outage we also look at the user impact. Figure 3.8 shows the percentage of impacted devices detected by M2MScan for all 14 outages. We also observe that for some outages an increase in p-value threshold have little effect on the percentage of impacted devices. For example, for outages 4 to 6 there is a small increase of approximately 1% user impact. Our analysis also reveals that outages with large user impact have large number of failed sites involved. For example outages 12 to 14 have more than 10% devices impacted, all of these outages have more than 10 failed sites.

We also visualize the histogram of start time and duration pairs in Figure 3.9. Here the xaxis represents the start time value denoted as i in our formulation and y-axis represents the window size denoted as W. The hottest points represent the outage start-time and duration pair for which maximum number of devices are classified as impacted. We observe that for a change in p-value from 0.01 to 0.02, the estimate of user impact increases by two times. Also, we do not observe significant shift in the outage start time duration values. However, for a p-value of 0.04 we observe a small change in the user impact estimate and large shift in the outage start-time and duration estimate. These observations highlight that small pvalues are more suited for estimating outage start times and durations. Also, larger p-value threshold provides better estimates of user impact. However, the downside of increasing the threshold value is an increase in number of false positives. The heatmap also shows that majority of devices were impacted with a similar duration and starting time. A geographical analysis of the outage reveals that the outage occurred in a region with a large number of cell sites located geographically close to each other. Therefore, in this case either the devices were able to switch to other nearby sites or were directly impacted. These example show the potential of M2MScan to identify the temporal impact of an outage.

To evaluate spatial impact of outages, we compute impact scores using the gaussian smoothing kernel. For each outage in a region R, we create 10 random subsets containing equal number of devices including both impacted and non-impacted devices. We use devices in 9 subsets denoted as d', to obtain the impact scores $G'_d(x, y) \forall d' \in D_R$ at the locations of devices in the 10-th subset. We then label each device in the 10-th subset as impacted or non-impacted based on the impact scores. For a given device in the 10-th subset we obtain |d'| number of scores. A device is impacted if the maximum score at its location belongs to another impacted device, otherwise we label the device as not-impacted. For 6 outages the impact scores correctly labeled 85% of the impacted devices. For remaining outages impact scores correctly labeled less than 10% of impacted devices. To investigate this variation we look at the geographical map of device and site locations. Our analysis reveals that due to terrain conditions and coverage planning done by cellular network, devices located within the outage region may connect to sites outside the outage region.

Next we discuss temporal characteristics of a special case of real site outage. Figure 3.10 shows the heat map plots generated from the frequency histograms. We observe that maximum number of devices are impacted with a starting time of 68 and a duration of 20. However, we also observe that a significant number of devices do not have this starting time and duration. For example, one particular group of devices has the same starting time with a duration of 5 time units. We also observe that devices belonging to the two different impact durations are disjoint set of devices. This shows that devices located in the same outage region can have different impact durations. To understand the varying impact durations we looked at the geographical locations of all qualified devices in the region. We found that the devices with smaller impact durations are located in the outer region of the outage and the devices with longer impact durations are located in the central region of the outage.

In summary, M2M devices enable us to analyze network outages at a higher granularity in both space and time. The impact duration and impact region of a given cell site outage can vary both in space and time. These variations are dependent on geographical terrain and cell site locations in the outage region. Besides obtaining an estimate of outage start time, duration, and user impact, M2MScan can be used to characterize network outages and their user impact.

3.7 Conclusion

In this chapter, we present our design and prototype of a M2MSourcing based monitoring system called M2MScan for a large-scale cellular network. We first identify a qualified set of M2M devices which are stationary and have consistent communication activity using data collected from a large-scale cellular provider in North America. We then use these qualified M2M devices to measure user impact of cell site outages. Our analysis reveals that impact of cell site outages is highly dynamic in space and time. We show that M2MScan is highly effective in estimating temporal impact of cell site outages. M2MScan is also useful in estimating percentage of user impacted due to a cell site outage. Spatial impact is highly dynamic due to geographical terrain and site locations, devices tend to handoffs to nearby working cell sites and remain active even if they are in the outage region To the best of our knowledge, large scale cellular network monitoring using M2M device at operational cellular networks has not been previously studied.

4 Optimizing Transit Route Selection for CDNs

4.1 Introduction

Content publishers usually rely on third-party Content Delivery Networks (CDNs) for efficiently delivering content to end users. A significant fraction of web content on the Internet is served by CDNs. Cisco estimates that the share of Internet video traffic served by CDNs will increase from 61% in 2015 to 73% by 2020 [12]. Two major considerations for CDNs are cost and performance of delivering content to end users. CDNs maintain copies of content at cache servers that are deployed at carefully selected geographical locations. CDNs also maintain multiple transit routes from cache servers to access ISPs (or "eyeball networks") which provide Internet connectivity to end users. When a user requests an object, the request is redirected to a nearby cache server containing the requested object. The cache server sends the object to the end user via one of multiple transit routes. Since the performance of transit routes changes over time, end-to-end performance achieved by a CDN is dependent on the choice of transit route [133]. Furthermore, the price of Internet transit also varies from one transit provider to another. Thus, a major challenge faced by CDNs is to develop a transit routing strategy to simultaneously optimize cost and performance.

The dynamic nature of transit pricing and performance makes it challenging to optimize the cost and performance tradeoff. There are thousands of eyeball ISPs which are reachable via different transit routes and different geographical locations. Each choice of transit route for a particular eyeball ISP and geographical location has distinct cost and performance characteristics, which makes the problem of developing a transit routing strategy challenging. Therefore, it is important for CDNs to carefully design and adopt a transit route selection strategy by analyzing the dynamically changing cost and performance tradeoffs.

To the best of our knowledge, the problem of optimal Internet transit routing for CDNs considering both cost and performance tradeoffs is not addressed in prior literature. Prior work on route selection has studied multi-homed access networks [44, 54, 149]. Unlike multi-homed access networks, CDNs have multi-homed servers at IXPs (via multiple transit providers) which provide explicit control over content routing. The state of the art for transit routing at IXPs by CDNs does not use a fully automated approach. CDN operators mostly configure routing strategies manually to optimize cost and performance. Due to performance dynamics across thousands of access ISPs and numerous transit routes, it is practically impossible for CDN operators to manually achieve an optimal tradeoff between cost and performance.

In this work, we present measurement, analysis, design, and evaluation of optimal transit route selection for CDNs. First, we present an approach to measure end-to-end performance across multiple transit routes simultaneously. We measure end-to-end performance as the delay between the CDN server located at an IXP and the end user. Second, we analyze the temporal and spatial dynamics of performance differences between multiple transit routes. Measuring performance differences across multiple transit routes allows us to characterize performance dynamics of Internet transit. Finally, we present our proposed approach for optimal transit routing. We formulate the transit route selection problem as using a multiattribute objective function that optimizes cost and performance simultaneously. Using linear programming, we obtain tradeoff curves between cost and performance for various routing strategies. Our results show that CDNs can achieve significant cost and performance benefits using our measurement and optimization approach. We face two key technical challenges in optimizing Internet transit routing. The first challenge is to obtain simultaneous performance measurements across multiple transit routes. This is necessary because performance at each transit route is dependent on multiple external factors. Performance differences between transit routes can vary due to congestion at the ISP-transit interconnections or congestion at intra-ISP links. Furthermore, performance can vary due to traffic engineering policies of customer ISPs. To solve this challenge, we capture the dynamics in performance differences by exploiting the multi-homing capability of CDN servers. Specifically, we implement a client-side performance measurement JavaScript that is embedded in client-requested web pages by the CDN. The JavaScript downloads multiple copies of a pixel tag simultaneously via multiple transit routes. This measurement methodology allows us to capture user perceived end-to-end performance via multiple transit routes. We measure performance of multiple transit routes simultaneously and focus on pixel tags whose download time is representative of Round Trip Time (RTT).

The second challenge is to obtain a transit routing strategy that provides an optimal tradeoff between cost and performance. For each IXP location, thousands of destination ISPs are reachable and for each destination ISP there are numerous transit routes. We solve this challenge by formulating the Internet transit routing problem as a multi-objective optimization problem. The objective function computes the utility of a particular routing strategy and optimizes it over all possible strategies. We describe utility as the sum of overall cost incurred to a CDN and performance weighted by a factor γ . By varying values of γ , we obtain various selection strategies. We obtain an optimal strategy by looking at the tradeoffs between cost and performance for different γ values. The ability to analyze and automatically control transit routing has significant cost and performance tradeoffs and accordingly choose an optimal routing strategy. We summarize our key contributions below.

• First, we characterize Internet transit performance from multiple IXP vantage points. Our comparative analysis of two transit providers reveals that one transit route significantly outperforms the other for more than 50% users at a European IXP and more than 30% of users at a North American IXP.

• Second, we propose an optimization approach that allows CDN operators to navigate cost and performance tradeoffs in transit routing through a control knob. Our results show that CDNs can reduce their transit costs on average by 57% without incurring any performance degradation.

4.2 Background

4.2.1 Architecture

CDN caching infrastructure consists of servers located at multiple geographically distributed locations. A content provider pushes copies of its content to the CDN. The CDN maintains copies of the content at geographically distributed cache servers. The client requests for content are typically directed to suitable CDN cache servers based on geographic proximity using DNS redirection or anycast [91].

There are two common server deployment strategies employed by most commercial CDNs [70]. CDNs either deploy servers inside many access ISPs that are closer to users (*enter-deep* strategy) or deploy servers at a few carefully chosen geographical locations (*bring-home* strategy). For example, Akamai has adopted the enter-deep deployment strategy, and has deployed 100,000+ servers across thousands of ASes [4, 132]. Several major content providers have also adopted the enter-deep deployment strategy. For example, Netflix's Open Connect Appliance (OCA) servers are deployed at ISPs delivering over 5Gbps in peak daily Netflix traffic [8]. Google Global Cache (GGC) servers are also installed inside large ISP networks [6]. In the bring-home strategy, CDNs deploy large clusters of servers at fewer sites and connect these sites with high-speed connectivity. Instead of deploying these clusters inside large ISPs, these CDNs strategically place their server clusters near Internet Exchange Points (IXPs). At IXPs, CDNs can interconnect with a large number of ISPs using peering or transit

[9, 96, 39]. According to a snapshot of the PeeringDB in August 2013, 76% of ASes use Open peering, 21% use Selective, and 3% use Restrictive [96]. For example, Limelight has 18,000+ servers at dozens of Points of Presence (POPs) around the world. CDNs interconnect with backbone transit ISPs, at IXPs to efficiently deliver content to end users.

Figure 4.1 provides an architectural overview of the bring-home CDN that we study in this work. As discussed earlier, we note that the CDN cache servers are located near major IXPs, where they interconnect with backbone *transit providers* (or transit ISPs).¹ CDNs buy transit services from multiple transit providers. CDNs can use one or simultaneously use multiple transit providers in order to minimize their transit costs and maximize performance for end users. Unlike enter-deep strategy, a bring-home CDN has more control over content delivery servers because cache servers are located at a small number of key geographical locations. However, as shown in prior literature [70], the CDN has to deal with larger end-to-end delay and higher transit costs as compared to enter-deep CDNs like Akamai. Therefore, it is crucial for bring-home CDNs, like the one discussed here, to carefully choose transit routes to optimize both performance and cost.



Figure 4.1 A CDN interconnects with multiple transit ISPs at IXPs

¹Note that the CDN can peer with an access ISP (or "eyeball network") to eliminate transit costs if the access ISP has presence at the IXP. However, small access ISPs typically do not have presence at multiple large IXPs [107]. Moreover, large access ISPs may not directly peer with the CDN due to the intricacies of peering [107]. In this chapter, unless stated otherwise, we restrict ourselves to transit routing for CDNs.

4.2.2 Internet Transit Dynamics

Pricing and performance of transit providers vary with respect to time and geographical location. Below, we provide an overview of both pricing and performance dynamics in the Internet transit market.

Pricing Dynamics. Internet transit prices have steadily decreased over the years due to technological advances and increased competition in the Internet transit market. Usage based and tiered pricing models are commonly used in the Internet transit market [137]. In the usage based pricing model, Internet transit is a metered service, i.e., transit providers charge their customers by measuring the amount of traffic sent or received during the billing period. Some transit providers may charge customers differently based on traffic volume and destination. In the tiered pricing model, transit providers charge customers based on geographical region, traffic commit levels, type of traffic i.e, on-net vs off-net, etc. [137]. The customers who commit higher bandwidths are able to negotiate lower per-Mbps costs as compared to the customers who commit lower bandwidths [108]. The most commonly used pricing scheme in the Internet transit market is called 95th-percentile pricing [131]. In this scheme, usage over a fixed billing period (typically one month) is measured on a megabit per second basis using the 95th percentile value. Unlike capped or fixed billing, where customers pay a fixed amount regardless of usage, 95th percentile charging is flexible. The service providers do not have to implement various charging policies and the customers pay only for what they utilize. Note that customers with bursty traffic are likely to pay higher costs than customers with consistent bandwidth utilization, even though overall traffic volume transferred by bursty customers may be less than the consistent ones. When considering the costs incurred due to effects of bursty traffic on traffic engineering policies, 95th percentile charging method balances the tradeoff between flexibility and the amount charged to customers.

Performance Dynamics. There are several factors that cause performance differences across transit routes. For instance, a transit route may simply be longer (more IP hops) than

others, resulting in consistently higher propagation delays. A CDN can easily identify such cases when a transit route is consistently worse than others. Transit performance is also affected by congestion at ISP-transit interconnections or congestion at intra-ISP links resulting in larger queueing delays and packet losses due to buffer overflows. The congestion can be temporary (e.g., during peak hours) or long-lasting indicting link under-provisioning. Such changes in transit performance are not in control of CDNs because contractual agreements between ISPs and changes in inter- and intra-domain routing policies are considered confidential information. From a CDN's perspective, it is important to continually monitor performance across different transit providers and choose transit routes accordingly to optimize end-to-end performance.

Overall, in addition to optimizing performance, CDNs also have to consider financial aspects of Internet transit routing. Different transit providers may charge differently, use different pricing models, and set up contractual agreements with/without performance SLAs. Thus, CDNs have to navigate the cost-performance tradeoff. A CDN can choose the cheapest transit route by sacrificing performance or can pay more for better performance. In this work, our goal is to understand the cost performance tradeoff from the perspective of CDNs.

4.3 Performance Measurements & Analysis

4.3.1 Measurement Methodology

In this section, we discuss our methodology to measure and analyze performance of transit providers. To measure the performance of multiple transit providers serving a particular ISP (e.g., AS in a specific geographical region), we utilize the multi-homing capabilities of the CDN's cache servers located at IXPs. Specifically, we embed a JavaScript in client-requested web pages to conduct active performance measurements. The client-side JavaScript generates HTTP requests to the multi-homed IXP server and downloads a pixel tag via different transit providers. The same pixel tag is downloaded simultaneously via multiple transit providers; this allows us to capture the performance differences of various transit routes simultaneously.

Figure 4.2 illustrates our measurement methodology. The JavaScript is embedded in the HTML pages served to end users. Once a client downloads the HTML page, the JavaScript executes in the background and sends an HTTP GET request for a pixel tag. To avoid additional delays incurred due to DNS lookups, we hard-code the IP addresses of the measurement server in the JavaScript. To avoid local cache hits, we add a nonce check in the client's HTTP request. This ensures that the pixel tag is served only from the multi-homed measurement server and not from the local browser's cache. We set the size of pixel tag to 10 kilobytes, which is less than the server's initial TCP congestion window. Thus, it takes approximately 2 RTTs to download the pixel tag. The JavaScript records the RTTs at the client side and periodically uploads the measurements to a database server.



Figure 4.2 Transit performance measurements

Our dataset consists of active measurements conducted from 19 measurement servers which are located at different IXPs. Each record in the dataset consists of time stamps indicating the measurement time, identifiers for transit providers, download time values obtained for each transit provider, client IP address, client AS number, and ISP name. In total, the dataset consists of 6 million entries which were recorded from more than 2 million IP addresses distributed across 16, 752 ASes. All user identifiers in the dataset are anonymized to protect the privacy of users. The measurement time span is distributed over a period of one year. Note that a measurement is recorded only when a client sends a request to download a web page. Therefore, the scope and size of our measurements depends on content popularity and user demographics. However, the CDN discussed in this study has thousands of clients covering a large number of ASes, which include most major residential broadband providers. We aggregate performance measurements on an hourly basis and use one hour as the time resolution for the rest of our analysis.

As shown in Figure 4.3(a), 19 measurement servers are located at geographically distributed IXPs and cover all major continents including North America, Europe, Asia, and Australia. We used the publicly available IP geolocation databases to locate host IP addresses. Figure 4.3(b) visualizes the IP address space of all host IP addresses in our data set. The plot divides the IP space by the first two most significant bytes as axes. The color indicates the number of records from each /16 IP block, where brighter colors represent more records and darker colors represent fewer records. The plot shows that our measurements covers a large chunk of the IP space. Some of the empty portions represent the space for reserved IP addresses. Other large empty portions correspond to unobserved (i.e., not served by the CDN servers), non-allocated, or inactive IP addresses.

Table 4.1 provides summary statistics of our measurements. Each row enumerates the total number of measurements, number of unique host IPs, number of unique ASes, and median download time for each transit provider. The three letter abbreviations in the first column denote various transit providers used by the CDN at various IXPs throughout the world. We note that Nippon Telegraph and Telephone (NTT) and Telia (TEL) provide the most coverage (in terms of number of IPs and ASes) to the CDN. Other transit providers



(a) Red dots represent the locations of measurement servers at IXPs and black dots represent the location of end users. (b) Each dot indicates count of measurements from a /16 IPv4 range block.

Figure 4.3 Deployment summary

are used at IXPs when NTT and TEL do not cover certain IXPs, or when NTT or TEL are temporarily unavailable. The median download times for NTT and TEL are 288 and 286 milliseconds, respectively.

Transit	#Records	#IPs	#ASes	Median		
Provider	x1000	x1000	ASes	$\mathbf{Time}(\mathbf{ms})$		
NTT	$5,\!435.7$	2,013.2	15,559	288		
TEL	5,217.5	1,923.5	$15,\!616$	286		
DTA	1,869.9	772.3	7,969	297		
PAC	552.9	236.8	1,910	488		
PCC	533.7	228.8	1,775	427		
AAP	192.3	814.3	189	241		

Table 4.1 Data set summary statistics

4.3.2 Analysis and Discussions

We first analyze the performance characteristics of different transit providers using our measurements. We are particularly interested in understanding *where* and *when* one transit provider outperforms another transit provider. To this end, we explore spatial (where) and temporal (when) variations in transit provider performance difference. In this work, we use the time to download the pixel tag for each transit route as a measure of its performance. If t_1 and t_2 denote the download time of the pixel tag via two transit providers, then let $\delta t = |t_2 - t_1|$ denote the performance difference in terms of download time. In the rest of this section we use the performance difference δt as performance metrics. Below, we focus our analysis to two major transit providers (NTT and TEL) across four popular POP locations (Paris, Madrid, San Jose, Chicago).

Spatial variations. Figure 4.4 plots the cumulative distribution of percentage performance difference between transit providers for users across four POP locations. The x-axis represents users and the y-axis represents the percentage performance difference between two transit providers for a given user. The plot indicates the portion of users who receive better performance from one transit provider versus the other transit provider. A positive percentage difference indicates that NTT has better performance, and a negative percentage difference indicates that TEL has better performance. Each curve is aggregated for all ASes that interconnect at the corresponding POP location. We observe that NTT provides significantly better performance than TEL for users on the left of the x-axis, while TEL outperforms NTT for users on the right side of the x-axis. For Madrid POP, we note that NTT outperforms TEL by 10% for approximately 40% of users and TEL outperforms NTT by 10% for approximately 10% of users. In sum, 50% users experience significant performance difference across two transit providers. The remaining 50% users would fare similarly on either transit provider. Using the same 10% performance difference threshold, we note that one of these transit providers outperforms the other for 55% users at Paris POP, 30% of users at Chicago POP, and 30% of users at San Jose POP. Our findings highlight that there is no outright best transit provider for all users and a careful choice of transit provider is necessary.

Temporal variations. To investigate temporal variations in performance differences between transit providers, we analyze our performance measurements over time. For each POP location, we again plot the distribution of percentage performance difference between transit providers for users in different one-month billing periods. Figure 4.5 plots the curves for four popular POP locations and for three different billing periods. We observe that for Paris and Madrid POPs, one transit provider outperforms the other for a vast majority of



Figure 4.4 Spatial performance variations for different POPs

users across all billing periods. For example, TEL outperforms NTT for up to 70% of users at the Paris POP and NTT outperforms TEL for up to 65% of users at the Madrid POP. While the overall trend remains the same for all POPs, we observe changes across different billing periods. For instance, the performance difference between NTT and TEL increases over time for San Jose POP. More specifically, NTT has equal or better performance as compared to TEL for most users in July. However, TEL's performance improves as compared to NTT's performance over the next billing months. By November, TEL outperforms NTT for more than 25% of users while NTT outperforms TEL for approximately 10% of users. Similar temporal variations in performance difference can be observed for all POPs. During certain consecutive months two transit providers may have similar performance difference characteristics, as a result we may not observe variations over consecutive months. Therefore, we choose different months for different POPs to show variations across different geographical locations at different times.

To understand finer-grained temporal variations in transit performance, we analyze the time series of percentage difference between transit providers across AS-POP pairs. We plot the hourly performance time series for the Telefonica-Madrid AS-POP pair in Figure 4.6. A positive value of $t_{\text{TEL}}-t_{\text{NTT}}$ indicates that NTT's performance is better than TEL's performance. For this particular AS-POP pair, we observe that NTT's performance is consistently better



Figure 4.5 Performance difference of transit providers for different billing periods

than TEL's performance. We plot the hourly performance time series for the Comcast-San Jose AS-POP pair in Figure 4.7. We observe that NTT's performance is generally better than TEL's performance during the first three weeks. However, the trend is quickly reversed in the fourth week. The reversal in performance seems to be due to improved performance of TEL (e.g., infrastructure upgrade) rather than degraded performance of NTT (e.g., traffic pattern shift). For rest of the measurement period, TEL consistently outperforms NTT. Figure 4.6 (b) and 4.7 (b) show the zoomed time series plots for the fourth week. We note diurnal variations for $t_{\text{TEL}} - t_{\text{NTT}}$, which indicate congestion during peak hours.



Figure 4.6 Difference in transit performance for AS-POP pair: Telefonica-Madrid



(a) Performance during 9 weeks (b) Zoomed in plot for week no. 4

Figure 4.7 Difference in transit performance for AS-POP pair: Comcast-San Jose

4.4 Problem Formulation

Due to the rich connectivity at IXPs, a CDN can choose from multiple (often dozens) of transit options to route traffic to end users. In particular, the traffic to an AS can be routed via one of many transit providers. As discussed earlier, the dynamics of cost and performance makes manual solution out of scope. We need to optimize both cost and performance for each destination AS. Our discussions with network engineers revealed that this optimization is often done manually on an hourly basis, and misconfigurations and performance issues are quite common. Due to a large number of ASes, network engineers typically do the manual optimization only for large ASes or the ones with poor performance [149, 2, 5]. Thus, we need automated solutions to the cost-performance optimization problem that can be configured for different cost-performance tradeoffs.

We formally define the dynamic cost-performance optimization problem for Internet transit selection using constraint programming. In a constraint programming approach an objective function is formulated with a set of constraints on variables of the objective function. Then a solution is obtained such that the variable values are within the specified constraints [53]. Constraint programming allows us to formulate the optimization problem in a form that can be used to obtain a tradeoff curve between cost and performance. More specifically, we design an objective function that minimizes a utility cost function. The function is a weighted sum of cost and performance of a transit provider selection strategy. Based on the tradeoff curve, CDNs can adapt transit provider selection to obtain the desired tradeoff.

For a POP P_j , consider a transit provider *i* that charges r^i units of cost to a CDN. Each POP serves multiple ASes and each AS can be served through multiple transit providers. For a AS-POP pair $\langle A_k, P_j \rangle$, let $b_{k,j}^i$ and $d_{k,j}^i$ respectively be the estimated bandwidth usage and performance through transit provider *i*. For usage based pricing r^i is a fixed amount that transit provider *i* charges on hourly basis. For 95th percentile pricing, we consider a billing period of one week and compute the 95th percentile of hourly bandwidth usage. We use this bandwidth usage to compute the transit costs. Specifically, for transit provider *i* the transit cost is $\sum_{k,j} r^i \times b_{k,j}^i$, where $b_{k,j}^i$ represents the 95th percentile value. We use number of records observed in each hour as an estimate of hourly bandwidth usage. The performance $d_{k,j}^i$ is measured in terms of download time in milliseconds. The CDN needs to select a transit provider for each AS-POP pair $\langle A_k, P_j \rangle$. Let $x_{k,j}^i$ be the optimization variable that assigns transit provider *i* to AS-POP pair $\langle k, j \rangle$, thus $x_{k,j}^i \in \{0, 1\}$. We formulate the minimization objective function as follows.

$$minimize \sum_{k,j} \sum_{i} r^{i} x^{i}_{k,j} b^{i}_{k,j} + \gamma x^{i}_{k,j} d^{i}_{k,j}.$$
(4.1)

The objective function minimizes the utility cost of assigning transit providers to POP-AS

pairs. The utility cost function consists of two terms. The first term computes the cost in dollars charged by the selected transit providers weighted by the bandwidth usage. The second term is the median download time observed at the transit provider multiplied by a tunable parameter γ . We choose to model performance as part of the objective function because modeling performance as a constraint may render the problem unsolvable as performance of all transit providers may not satisfy the constraint. It is also possible that multiple transit providers may satisfy the constraint resulting in selecting the transit provider with poorer performance. The CDN operator can vary γ to obtain a desired tradeoff between cost and performance. Essentially, overall utility is the weighted sum of cost and performance. Smaller values of γ push the optimizer towards a minimum cost solution whereas larger values of γ push towards a better performance solution. The objective function is subject to the following constraints:

$$\sum_{i} x_{k,j}^{i} = 1 \forall k, j \tag{4.2}$$

$$x_{k,j}^i \in \{0,1\} \tag{4.3}$$

$$\sum_{k,j} x^i_{k,j} b^i_{k,j} \le C^i \tag{4.4}$$

Here C^i is the capacity of transit provider *i*. The first constraint ensures that only one transit provider is assigned to a POP-AS pair. The second constraint ensures that assignments are integral. The third constraint sets the limits on maximum bandwidth utilization by transit providers.

Note that the second constraint for ensuring integral assignments results in a combinatorial optimization problem, where the optimal solution can be obtained by evaluating the objective function over all possible assignments. We can utilize openly available Mixed-Integer Programming (MIP) solvers to obtain the optimal assignment. However, in practice each POP connects to thousands of ASes through multiple transit providers. Since Integer Programming (IP) is NP-Complete, it is not feasible to solve it for large input sizes [83]. We provide the proof of its NP completeness by reduction from the boolean satisfiability problem 3-SAT. The boolean satisfiability problem has been proven to be NP-complete using the Cook-Levin theorem. By reducing 3-SAT to IP we prove that IP is NP-Complete. A SAT instance is composed of variables and clauses. Variables in the SAT instance are boolean and their values should be such that the clause must result in a boolean TRUE.

Proof. For the 3-SAT problem consider the boolean variables $b_1, b_2, ..., b_n$. Consider a similar IP instance with the same number of integer variables denoted as $x_1, x_2, ..., x_n$. Values of each integer variable is constrained by the inequality:

$$0 \ll x_i \ll 1 \forall i \tag{4.5}$$

For any 3-SAT clause, there is a corresponding constraint. Consider a 3-SAT clause:

$$b_1 \vee \bar{b_2} \vee b_3 \tag{4.6}$$

The corresponding constraint is given by the following inequality:

$$x_1 + (1 - x_2) + x_3 \ge 1 \tag{4.7}$$

This reduction can be done in polynomial time.

Thus, we relax the constraint to allow fractional assignments i.e., $x_{k,j}^i \in [0,1]$. This relaxation allows us to obtain optimal solution with fractional assignments in linear time using standard Linear Programming (LP). Following lemmas state the relationships between solution obtained through LP relaxation and Integer LP (ILP) [122, 138].

Lemma 4.4.1. Let S_{LP} be set of feasible solutions in the LP and let S_{ILP} be set of feasible solutions in ILP. All feasible solutions in ILP are also feasible solutions in LP; i.e., $S_{ILP} \subset$ S_{LP} . For the minimization problem, solution obtained through relaxed LP has smaller value than the solution obtained through original ILP i.e., $\min(LP) \leq \min(ILP)$.

An optimal solution in LP may be found at the boundaries of the convex hull formed by the linear constraints. However, the feasible solutions in ILP are given by a set of points inside the convex hull formed by the linear constraints. These set of points do not form a convex set. Hence, value of the optimal solution in a minimization problem in LP is less than or equal to the value of an ILP optimal solution. Following the inequality in Lemma 4.4.1, we next state the condition under which both optimal solutions for LP and ILP are equal.

Lemma 4.4.2. Let X^o denote the optimal solution of the original ILP problem. Let X^l denote the optimal solution obtained from LP. If the optimization variable $x_{i,j}^k$ of the LP takes integral values in X^l , then it is also an optimal solution for ILP and hence the optimal solution for the original problem; i.e, $X^l = X^o$.

Optimal solution of a LP as stated in the lemma generally does not exist. Several heuristics have been proposed in the literature to obtain an integral solution from the relaxed LP [43]. In this work, we use the relaxed version of the minimization problem. For our empirical evaluation, we obtain a small percentage of fractional values in the solution to the relaxed version of the original problem. The fractional assignments can be dealt heuristically through randomized rounding or greedy assignments [116]. Another approach to deal with fractional assignments is to employ fractional routing, which can be realized by hash-based splitting or through multi-homing agents [25, 104].

4.5 Results

In this section, we discuss the cost and performance benefits of our proposed optimization approach for transit selection by CDNs. For pricing, we use usage-based and 95th percentile pricing models to compute costs incurred by the CDN in using various transit providers. On an hourly basis, we compute the median bandwidth usage for each AS-POP pair. For usagebased pricing, we study transit provider selection for two different scenarios; first, when all transit providers charge equal pricing rates (per Mbps), and second when all transit providers charge different pricing rates. The optimization framework works on hourly basis,



Figure 4.8 Cost-performance tradeoff curves for γ ranging from 0 to 5.

and the optimization works for each PoP independently over all ASes and transit providers. Therefore, differences in pricing for different POP locations (which is very common) does not impact transit providers across POPs. For 95th percentile charging, we use one week as the billing period in our experiments. 95th percentile charging is based on 95th percentile of all measurements over the billing period. Therefore, the optimization framework works on weekly basis under this charging model. For performance, we use download time of active measurements over a year as input to the optimization problem. For bandwidth, we use number of records observed in each hour as an estimate of bandwidth usage in that hour. As number of records are dependent on clients requesting the pixel tags, larger number of requests indicate greater bandwidth usage, therefore, we use number of records as an estimate for bandwidth usage. Specifically, we compute the median download time in every hour for all transit providers for each POP-AS pair. For a given transit provider, our system records several measurements for each AS-POP pair during a particular hour. We use the median value of these measurements in our experiments. We use an open source implementation of Embedded Conic Solver (ECOS) for solving the optimization problem. Given the solver output, we can compute the performance and cost of traffic via transit providers. While our

optimization framework is scalable for more than two transit providers, we limit our analysis to two transit providers (namely NTT and TEL) for simplicity.

4.5.1 Tradeoff analysis

We first study the tradeoffs between cost and performance in the Internet transit selection. By varying γ , we obtain a tradeoff curve between cost and performance for each POP. For each hour, we compute the cost and download time using the first and second terms of our objective function, respectively. We repeat this process for different γ values and obtain a tradeoff curve between total cost and average download time per AS. Each point on the tradeoff curve represents a transit provider selection strategy for all ASes served by the POP. We solve the optimization problem every hour and use average to obtain a tradeoff curve.

Figure 4.8 plots the cost-performance tradeoff curves for selection between two transit providers at four POP locations. We plot the tradeoff curves for usage-based pricing models and 95th percentile pricing models in Figures 4.8(a) and (b), respectively. The x-axis is the average download time per AS (in milliseconds) and the y-axis represents the incurred transit costs (in dollars). Each point on the curve represents the average cost and performance over a period of one month for a given γ value. To plot the tradeoff curves, we vary the values of γ from 0 to 5 with an increment of 0.1. For smaller values of γ , we obtain solutions with lower costs but worse performance. For $\gamma = 0$, we get the lowest cost solution that selects transit providers with minimum costs irrespective of their performance. As we increase the value of γ , we give higher weight to performance and we obtain higher cost solutions with better performance. For $\gamma = 5$, we get the best performance solution that always selects the transit providers with the lowest download time. The pairs of dots on each curve represent the minimum cost solution for $\gamma = 0$ and best performance solution for $\gamma = 5$. The lines are extrapolated to show solutions that can be obtained for values of γ less than 0 and greater than 5. Choosing larger values of γ produces solutions on the vertical portion of the tradeoff curve. This means that we always obtain the same best performance solution regardless of the cost. Therefore, we only choose values of γ that produces meaningful tradeoffs between cost and performance.

Comparing Figures 4.8(a) and (b), we note that the CDN ends up paying more under the 95th percentile pricing model. For Paris and Madrid POPs, we observe smooth tradeoff curves, indicating that the CDN can select among a number of strategies by varying γ around the knee of the tradeoff curves. We also observe that regardless of the pricing model, the CDN can save more than 50% on cost with a performance degradation of less than 40 milliseconds. For Chicago and San Jose POPs, we observe sharp knees of the tradeoff curves which indicate that slightly changing γ values has a substantial impact on cost and performance. For these POPs, the CDN can save on average 30% on cost with a performance degradation of less than 20 milliseconds. For large values of γ , the CDN may end up paying a lot more even though there is much room for savings without significantly degrading the performance. Thus, there is very limited flexibility for CDNs in terms of optimal transit provider selection. In summary, the tradeoff curves under the usage based pricing model and the 95th percentile charging model show that the CDN can reduce transit cost by 57% and 35% respectively, without significant degradation in performance.

4.5.2 Transit selection analysis

Recall that transit provider selections are outputs of our optimization framework, i.e., the values of the target variable $x_{i,j}^k$ provides selection results. We analyze transit provider selections for three different usage-based pricing scenarios: (1) Both transit providers charge equal rate, (2) NTT charges higher rate than TEL, and (3) TEL charges higher rate than NTT. For equal pricing scenario, the optimization framework provides the performance-optimal solution. For unequal pricing scenarios, the optimization framework selects the best transit provider in terms of both cost and performance.

Figure 4.9 plots the timeseries of performance difference between TEL and NTT for an example POP-AS pair. The primary y-axis (left side) shows the difference in performance

	Paris		Madrid		San Jose		Chicago					
	NTT	TEL	Frac	NTT	TEL	Frac	NTT	TEL	Frac	NTT	TEL	Frac
NTT and TEL charges equal price	59.5	75.3	2.1	75.3	62.7	1.8	72.3	67.6	7.8	63.6	79.6	9.9
NTT charges higher price	40.0	87.4	0.1	59.6	78.1	0.0	41.7	89.7	0.1	41.3	90.5	0.0
TEL charges higher price	78.4	57.3	0.86	87.3	38.8	0.0	91.6	34.3	0.2	84.2	52.9	0.0

Table 4.2 Summary of assignments for different pricing rates and POP locations

measurements between the two transit providers. The secondary y-axis (right side) shows the output selection values for transit provider TEL, where 1 indicates TEL is selected and 0 indicates NTT is selected. Blue dots represent transit provider selections (right y-axis). Figure 4.9(a) shows selections when both transit providers charge equal rates. We observe that during the first four weeks TEL performs worse than NTT, as a result TEL is not selected. For the later half, the performance difference becomes negative indicating that TEL is performing better than NTT, thus TEL is selected during the later half. Figures 4.9(b) and (c) show selections when NTT and TEL charge different rates. In both cases, we selected a γ value that provides the best tradeoff between performance and cost, i.e., the knee of the tradeoff curve. When NTT charges more, TEL is mostly selected even when TEL has worse performance. Whereas when TEL charges more, we see selections for both NTT and TEL. This shows the impact of performance difference and γ values on transit provider selection. In the later half, performing transit provider. Therefore, even though TEL is charging more than NTT, TEL is selected because performance improvement trumps additional cost.

In order to systematically quantify the impact of different pricing on transit route selection, we compare selections for varying pricing rates. Figure 4.10 plots the percentage of selections obtained from the optimization framework for varying price ratio between NTT and TEL. The y-axis shows the fraction of total number of hours in the input data. Each pair of symmetric red/blue lines correspond to a given γ value. The fraction of NTT selections are shown in blue and the fraction of TEL selections are shown in red and are symmetric to the corresponding blue line. Price ratio is calculated as $\frac{\text{TEL } price}{\text{NTT } price}$. The plot shows that when TEL is 10 times cheaper than NTT, TEL has larger percentage of selections. Furthermore, when TEL is 10 times



(c) TEL charges higher pricing rate

Figure 4.9 Transit provider selection for different pricing scenarios.

expensive than NTT, selections for NTT reach more than 90% for smaller γ values and more than 85% for larger γ values. Overall, we observe that as γ increases the fraction of NTT selections also increase regardless of performance. This finding indicates that overall NTT is favorable in terms of performance.

We also observe some fractional transit assignments by the optimization framework. We summarize the integral and fractional assignments as follows. Let δ_{tp} denote total number of hours during which transit provider tp is better. Let Sel_{tp} denote the number of hours transit provider tp is better and it is selected i.e. $x_{i,j}^k = 1$ for i = tp. Then $\frac{\delta_{tp}}{Sel_{tp}}$ is the fraction of



Figure 4.10 Transit provider selection for varying price ratio: <u>TEL price</u>.

hours during which the performance difference was large enough to select transit provider tp. For cases when we have non-zero values of gamma and unequal transit provider pricing rates, the magnitude of performance difference has to be large enough to affect transit route selection. In summary, for the equal pricing scenario, NTT and TEL were selected integrally 72.3% and 67.6% of the time with fractional assignments 7.8% of the time. When NTT is charging more, these percentages are 41.7%, 89.7%, and 0.1%. When TEL is charging more, these percentages are 91.6%, 34.3%, and 0.2%. We summarize the transit provider selections for four different POP locations in Table 4.2. We observe that pricing rates have a major impact on transit provider selections for all POP locations. We also observe that the percentage of fractional assignments is larger when under both transit providers charge equal pricing rates. Since the percentage of fractional assignments is generally small, we deem it unnecessary to utilize heuristics to obtain integral solutions in polynomial time.

We next analyze the effect of varying the weight factor γ on transit provider selections. We plot the normalized frequency histograms of TEL and NTT selections for performance difference values. Figures 4.11 show the histograms for the three different charging scenarios at two POP locations. Figures 4.11(a) shows that under equal cost changing γ values do not affect transit provider selections. We observe that the percentage of TEL selections is more than NTT



Figure 4.11 Effect of γ values on transit provider selection (Paris)



Figure 4.12 Analysis of transit provider bandwidth usage

selections when TEL provides better performance (negative x-axis). Similarly, the percentage of NTT selections is more than TEL selections when NTT provides better performance (positive x-axis). In Figure 4.11(b), we observe no change in percentage of TEL selections when TEL provides better performance; i.e, when TEL is better and cheaper, TEL is always selected irrespective of the choice of γ . However, we observe that as γ increases the fraction of NTT selections also increases when NTT provides better performance. This pattern indicates that when NTT is better and expensive, increase in γ value gives more weight to performance, as a result, the percentage of NTT selections increases. We observe similar patterns in Figures 4.11(c). In summary, a transit provider's selection does not significantly change for varying γ values if it provides better performance and is cheaper.

We further analyze the timeseries of bandwidth usage for competing transit providers in Figure 4.12. Note that bandwidth usage is calculated using transit provider selection information and traffic volume for each AS-POP pair. The \uparrow symbol denotes the more expensive transit provider and the \downarrow symbol denotes the cheaper transit provider. No arrows indicate that both transit providers have equal costs. For these AS-POP pairs, we observe that under equal costs the best performing transit provider is selected most of the time. We also observe some cases where both transit providers are being used indicating fractional assignments. The timeseries also show short-term and long-terms traffic shifts. In Figure 4.12, we note long-term traffic shifts when transit provider TEL is expensive. We observe that most of the traffic is routed through TEL for the first three weeks. During the later half of the duration we observe that most of the traffic is routed through NTT which is cheaper.



(a) Solution from pre-selected γ values (b) Solution obtained from Algorithm 1

Figure 4.13 Pareto fronts for different hours.

4.6 Pareto Front Generation

In this section, we extend our tradeoff results by generating sets of pareto-optimal solutions on hourly basis. The aggregate tradeoff curves in Figure 4.8 are obtained by taking average of pareto fronts belonging to several hours. For any particular hour, a pareto front is generated by using a set of pre-selected values of the weighting factor γ . CDN operators need to perform transit route selection on hourly basis. Therefore, they have to navigate the cost and performance tradeoffs on hourly basis. Operators can analyze the pareto front for any hour and navigate tradeoffs to choose a solution from the set of pareto solutions.

A pre-selected set of γ values is not enough to navigate the cost performance tradeoffs. There could be several unexplored regions over the pareto front where better tradeoffs can be achieved. Figure 4.13(a) shows the pareto fronts obtained for different hours using the preselected set of γ values. The figure shows that for some hours the pareto optimal solutions are not evenly distributed over the pareto front. In addition, the solutions are far apart and the tradeoff characteristics of the pareto front are not completely visible.

In order to obtain a well distributed set of solutions on the pareto front, the set of γ values has to be selected carefully. To find solutions in the unexplored regions of the pareto front, we use a recursive approach proposed by Kim *et al.* in [88]. The approach explores pareto front by recursively selecting a new set of γ values for the unexplored regions. Algorithm 2 describes the recursive method for Pareto front generation. An initial set of Pareto solutions \mathbb{P}_0 corresponding to a pre-specified set of γ values is used as an input to the method. First, Euclidean distance D_i of all segments formed by points in \mathbb{P} is calculated. If the length of segment i is less than a threshold ϵ , then this means that the two solutions (p_i, p_{i+1}) are nearly overlapping. For solutions very close to each other on the Pareto front, we do not see any significant cost-performance tradeoffs, therefore we do not perform any operations on the unexplored solution space between the two points. For a segment i formed by Pareto solutions (p_i, p_{i+1}) , and with length greater than ϵ , we obtain a set of γ values \mathbb{G}_i . All γ values in \mathbb{G}_i range from the γ value corresponding to p_i to the γ value corresponding to p_{i+1} . Next, we obtain a set of Pareto solutions \mathbb{P}_i , for all γ values in \mathbb{G}_i . Finally, we use the Pareto solutions in \mathbb{P}_i to recursively obtain solutions on the unexplored regions of the Pareto front. Figure 4.13(b) shows the pareto fronts obtained through the recursive approach. In comparison to Figure 4.13(a), we observe that several solutions are obtained in the unexplored regions of the pareto fronts. For example, pareto front belonging to Paris at 1 am has new solutions in the region between normalized download time of 0.6 to 0.9. We compare of pareto front

Algorithm 2 Generating pareto fronts recursively

Input: (1) Initial set of γ values: $\mathbb{G}_0 = \{\gamma_1, \gamma_2, ..., \gamma_n\}$

- (2) Initial set of Pareto solutions: $\mathbb{P}_0 = \{p_1, p_2, ..., p_n\}$
- (3) Minimum step size ϵ

Output: Final set of Pareto solutions: \mathbb{P}

1 Function $ParetoFront(\mathbb{G}_0,\mathbb{P}_0,\epsilon)$

I.

solutions obtained before and after using the recursive approach using variance of the length of segments formed by solutions on the pareto front. The solutions in Figure 4.13(b) have approximately three order of magnitude lower variance. This shows that cost-performance tradeoff is more flexible for CDN operators as they can choose from a variety of solutions which are evenly distributed on the pareto front.

4.7 Related Work

There is prior work on route selection from the perspective of stub ISPs, edge networks, and enterprise networks. Several studies have measured and analyzed performance and pricing aspects of such multi-homed networks. In contrast, this chapter studies transit route selection at IXPs by CDNs.

Multi-homing: Akella et al. presented a large scale measurement based performance analysis of multihoming [25]. The authors evaluated multihoming performance from the perspectives of enterprises and content providers, and found that multihoming improves performance. In [44], Dhamdhere et al. proposed a two step methodology for optimizing cost and performance in multi-homing. Tao et al. also explored the performance benefits of path switching in multi-homing scenarios and overlay network scenarios [134]. In [56], Yang et al. proposed smart routing algorithms to optimize cost and performance for multihomed enterprises and stub ISPs. The customer ISPs are assumed to be multihomed to a set of ISPs and they share ISP links for multihoming purposes. The proposed algorithms dynamically assign traffic to achieve optimal cost and performance. In [54] the authors showed that route control systems in multihomed networks can cause traffic oscillations between available paths which have adverse affects on performance. The authors used bandwidth information and randomization in their algorithms to avoid oscillations. In [140], Wang et al. studied monetary aspects of multihoming. The authors proposed a dynamic programming algorithm to obtain a subset of ISPs for multihoming purposes so that the cost is minimized for the They also study dynamics of ISP pricing strategies in response to customer customer. subscriptions. More recently, in [149] and [104], the authors proposed measurement and optimization frameworks for service provider selection for content delivery from content providers to its users.

As compared to prior work on multi-homing, our work is different in two major dimensions: First, we focus on multi-homed CDN servers at IXPs, which offer a large number of possible interconnections between CDNs and ISPs. Prior work on multi-homing assumes the multihomed network to be an edge network, whereas in our case we deal with multi-homed CDN servers at IXPs. Second, our light-weight measurement methodology is unique as it allows us to measure performance across multiple transit routes simultaneously at a large scale.

Transit pricing: Pricing in the internet transit has been studied from two different perspectives. First, from the transit ISP perspective, where transit ISPs aim to maximize their profit while satisfying service quality and customer traffic demands. Prior work has studied the impact of adopting various pricing strategies on ISP profits such as time-dependent pricing, tiered pricing, and pricing differentiation [77, 87, 62]. In particular, Valancius *et al.* studied the affect of destination based tiered pricing on ISP profit [137]. The authors developed demand and cost models using real-world traffic data and showed the impact of various traffic bundling strategies on ISP profits. In contrast, we study pricing from the transit ISP customer's perspective, where CDNs aim to minimize transit costs while achieving best performance. For transit costs [130, 60, 26, 37]. Our work focuses on cost reduction by selecting best available transit route which also provides best performance.

Server deployment: Some prior work focuses on optimizing server deployments by CDNs. In [115], the authors study the problem of online placement of servers to minimize cost in terms of latency, hop count, and economic cost. In [51], the authors propose a system that enables CDN-ISP collaboration leading to informed server placement and end-user to server assignment. In [65], the authors formulate and solve the cache deployment optimization problem to minimize CDN deployment costs while maintaining end-user performance.

4.8 Conclusions

In this chapter, we studied the problem of optimal Internet transit selection from the perspective of CDNs. To the best of our knowledge, transit route selection at IXPs by CDNs is
not studied in prior literature. We make the following key contributions in this chapter. First, we propose a method to conduct simultaneous performance measurements across multiple transit routes which are maintained by CDNs for delivering content to access ISPs. Second, we provide spatio-temporal characterization of performance differences observed across multiple transit routes. Our findings highlight that there is not an outright best transit provider. Third, we formulate the optimal transit route selection problem as an optimization problem and analyze tradeoff curves and resulting selection strategies for different pricing of transit routes. We observe sharp knees in tradeoff curves for some geographical regions, indicating that there is very limited flexibility for CDNs in terms of choosing a suitable transit provider. Without an optimal selection strategy, CDNs may end up sacrificing performance and/or cost. Using our proposed approach, CDNs can reduce transit costs on average by 57% without sacrificing performance.

5 Measurement and Analysis of Adult Traffic

5.1 Introduction

5.1.1 Background

As the saying goes: "The Internet is for porn" [30]. While it is difficult to estimate how much porn content is available on the Internet [142], there are several widely varying estimates available. According to one estimate, there are at least 4 million adult websites on the Internet, which constitute approximately 12% of all websites [13]. It has also been reported that adult websites have more monthly unique visitors than Netflix, Twitter, and Amazon combined [14, 109]. Furthermore, multiple websites in the Alexa's top-500 global list serve adult content [1]. Moreover, a recent measurement study of a tier-1 ISP reported that at least 15% of all mobile video traffic in the Unites States is from adult content providers [48]. Overall, these statistics indicate that online adult content attracts a large number of users and accounts for a substantial fraction of the global Internet traffic. However, despite its significant volume, little is known about the makeup and characteristics of online adult traffic. The understanding of online adult traffic is important for optimizing its content delivery, which involves complex interactions between content delivery networks and ISPs.

5.1.2 Limitation of Prior Art

There is little prior work on dedicated analysis of online adult websites. Only recently, Tyson *et al.* [135, 136] studied behavioral aspects of two popular adult sites: YouPorn and PornHub. These studies reveal several unique characteristics of adult content such as the elasticity of adult content consumption – users consume whatever they find on the frontpages of adult websites. However, the utility of these studies is limited due to three main reasons: (1) they rely on website data obtained by crawling, which is limited in terms of both temporal coverage and granularity; and (2) their analysis cannot distinguish among users because they rely on aggregate view counts of adult videos; and (3) the focus of these studies is on the behavioral and demographic aspects of two specific adult websites, and whether their findings are representative of other adult websites is unclear.

5.1.3 Main Findings

In this chapter, we conduct the first large-scale measurement study of online adult traffic using HTTP logs collected from a major commercial content delivery network. The weeklong HTTP logs include traffic from several dozen major adult websites and their users in four different continents. Overall, our HTTP logs account for approximately 323 terabytes worth of traffic from 80 million users. Based on these traces, we present some detailed characteristics of five popular adult websites. We choose these websites on the basis of their popularity, e.g., several of these adult websites have been ranked in the global Alexa top-500 list [1]. This selection represents a broad variety of adult websites, ranging from adult video services, adult image sharing services, to adult social networking services.

We provide an in-depth analysis of online adult traffic including its *aggregate*, *content*, and *user* dynamics. Below, we summarize our key findings and their implications on adult content delivery.

• Aggregate Analysis: Adult traffic primarily comprises of video and image multimedia content. For many popular adult websites, up to 99% traffic volume consists of video

and image content. While the majority of users access adult websites from desktop, smartphones and other mobile devices account for a non-trivial fraction of visitors.

- Content Analysis: Adult content has widely varying sizes: images are generally less than 1 megabyte and videos are on the order of tens of megabytes. Content popularity distributions exhibit the expected skewness. The temporal access patterns of adult websites are unique and different from the typical diurnal access patterns of traditional web content. Our clustering analysis of content popularity reveals groups of objects with diurnal, long-lived, and short-lived temporal access patterns.
- User Analysis: User engagement with adult websites is shorter than non-adult websites. However, adult content can be addictive, i.e., some users repeatedly access certain content. For example, at least 10% of video objects have more than 10 requests per unique user.
- Implications: Our findings have implications on adult website design and content delivery infrastructure management. For example, a vast majority of users do not visit adult websites on smartphones. This finding highlights the need for adult websites to improve their web interfaces and content delivery strategies for mobile devices. As another example, due to their unique diurnal access patterns, it is important to separately account for adult traffic in the traffic forecasting models and network resource allocation. We also find that adult content providers cannot rely on browser cache to store locally popular content because of prevalent use of incognito/private web browsing. Adult content providers can instead optimize content caching performance by customized networked cache configuration. For example, content delivery networks can improve performance and reduce network traffic by pushing copies of popular adult objects to locations closer to their end-users.

5.2 Related Work

Despite a large number of studies on general web traffic analysis (e.g., [103, 28, 148, 145, 112, 126]), little is known about the makeup and characteristics of online adult traffic. To fill this gap, this chapter presents the first in-depth and large-scale characterization of online adult traffic. To provide some context for our study, we review prominent related work below.

Wondracek *et al.* studied the economic and security issues of the online adult industry [144]. In their study, the authors used manual inspection and automated crawling to investigate the characteristics of adult websites. They classified adult websites based on their functionality into the following categories: paysites, link collections, search engines, domain redirector services, keyword-based redirectors, and traffic brokers. They also created two adult web sites from scratch for traffic profiling and vulnerability assessment. They concluded that several prevalent practices in online adult industry are questionable and can be used to conduct malicious activities. This work provides a broad understanding of the economic and security issues of the online adult industry. In contrast, we analyze HTTP access logs of several popular adult websites to understand their content and user traffic patterns.

More recently, Tyson *et al.* conducted measurement studies of two Web 2.0 adult websites YouPorn and PornHub [135, 136]. In [135], the authors crawled YouPorn to understand user interactions with the website. This measurement study shows that the fraction of user generated content on YouPorn is much smaller as compared to non-adult video websites like YouTube and Vimeo. However, they found that YouPorn has more average views per video than other non-adult video websites. The authors concluded that the video viewing behavior on YouPorn is dependent on two main factors: front page browsing patterns and the number of categories assigned to a particular video. They found that videos with most views are located on the front page of the website and the number of views a video receives is correlated with the number of categories assigned to it.

In [136], Tyson *et al.* crawled PornHub, an adult social networking website, to analyze the demographic makeup of users. The study shows that a majority of profiles on PornHub belong to young males, however female profiles are more popular in terms of receiving more comments and profile views. The authors also found that active profiles have larger social groups and there is positive correlation between content uploaded by a profile and its number of connections. The data analyzed in these studies is collected by periodic crawling of these two adult websites, which is generally limited to meta-data like view counts, ratings, user profile information. In contrast, our work is based on analysis of detailed HTTP access logs of multiple adult websites, which allows us to track individual user content requests at a fine-grained timescale.

Some prior web traffic measurement studies have reported basic statistics of adult traffic in the context of overall traffic makeup. For example, Du *et al.* studied HTTP traffic from Internet kiosks from two developing countries and reported that adult content accounts for less than 1% of total traffic volume [45]. Erman *et al.* studied video traffic in a large 3G cellular network and reported that adult content accounts for approximately 15% of total mobile video traffic [48]. While these studies are useful, they do not specifically analyze adult traffic in terms of its unique characteristics as compared to non-adult traffic. To the best of our knowledge, our work provides the first analysis of online adult traffic at scale.

5.3 Measurement & Analysis

Most major adult and non-adult content publishers use third-party content delivery networks (CDNs) to efficiently deliver their content to end-users. According to recent estimates, a significant fraction of Internet traffic is served by CDNs [93]. For instance, Akamai alone delivers between 15-30% of all web traffic [3]. A CDN operator typically places content at multiple geographically distributed data centers. A user's request for content (such as a web page, an image, or a video file) is redirected to the closest data center via DNS redirection, anycast, or other CDN-specific methods [111, 52].

For this study, we collected HTTP access logs from multiple data centers of a major

commercial CDN for the duration of one week. The HTTP logs include traffic from several dozen major adult websites and their users in four different continents. Overall, our HTTP logs account for more than 323 terabytes worth of traffic from 80 million users. All personally identifiable information in the HTTP logs (e.g., IP addresses) is anonymized to protect the privacy of end users without affecting the usefulness of our analysis. Each record in our trace includes information about an HTTP request, containing publisher identifier, hashed URL, object file type, object size in bytes, user agent, and the timestamp when the request was received. We use the user agent field to distinguish between different device types, operating systems, and web browsers [49]. Each record in our trace also includes information about the corresponding HTTP response sent by the CDN server, containing the cache status for the requested object. A HIT value indicates that the requested object was found in the CDN cache and a MISS value indicates that the file does not exist in the CDN cache. We use the HTTP response codes and cache status information to measure caching performance of proprietary CDN caching algorithms.

Through an extensive manual analysis of publisher identifiers, we separated adult content publishers from the rest (dubbed as "non-adult"). We select five popular adult websites in our data set for further in-depth analysis. The names of these websites are anonymized due to business confidentiality agreements. For adult websites, two websites primarily serve adult video content (termed V-1 and V-2), two websites provide image-heavy adult content (termed P-1 and P-2), and one is an adult social networking website (termed S-1). In this section, we analyze aggregate statistics of adult websites (e.g., content composition), content dynamics (e.g., popularity, new content injection), and user dynamics (e.g., session length, addiction to adult content).



Figure 5.1 Content composition of five adult websites.

5.3.1 Aggregate Analysis

5.3.1.1 Content Composition

We first analyze the content composition of adult websites on CDN servers. To this end, we categorize objects based on their file types into video (e.g., FLV, MP4, MPG, AVI, WMV), image (e.g., JPG, PNG, GIF, TIFF, BMP), and other (e.g., text, audio, HTML, CSS, XML, JS). From Figure 5.1, we note that V-1 primarily stores video objects on the CDN servers, e.g., 98% of all objects are videos. V-1 uses 6.6K objects, V-2 uses 55.6K objects, P-1 uses 16.3K objects, P-2 uses 29.6K objects, and S-1 uses 22.9K objects. We breakdown content into 3 categories: (1) video, (2) image, and (3) other. Other category includes objects that are not classified as video or image. In contrast, V-2 stores a mix of image (84%) and video (15%) objects. V-2 uses a large number of GIF images to show a video summary when users hover the cursor over the video. P-1, P-2, and S-1 store images (99%) on CDN servers.

5.3.1.2 Traffic Composition

We next analyze the traffic composition of adult websites in terms of the request count and request size during the data collection period. Request count is the total number of requests received from website visitors for all objects. Request size is the total size of objects requested by website visitors for all objects. We again breakdown content across video, image, and other



Figure 5.2 Traffic composition of five adult websites.

categories respectively. Figure 5.2(a) shows the traffic composition distributions with respect to their request count. We observe that the majority of traffic on adult websites consists of video and images. Only V-1 has significantly more video traffic than other content types - V-1 traffic includes 3.1M requests for video objects. V-2 has smaller percentage of video content traffic as compared to images or other categories. For V-2, 359K requests are for video content whereas 657K requests are for image content. For P-1 and P2, 719K and 175K requests are for images, respectively. For S-1, 231K requests are for images. Figure 5.2(b) shows the traffic composition distributions with respect to their request size. In contrast to Figure 5.2(a), we note that video content accounts for disproportionately more traffic volume. Since video files are significantly larger than image files, videos tend to dominate the traffic in terms of byte volume. Video traffic in V-1 alone accounts for 258 gigabytes worth of traffic. This traffic mix is composed of mostly multimedia content and it is representative of popular free (ad-driven) and subscription based adult websites [144]. Our findings highlight that the adult content publishers and the respective CDNs need to design and provision their infrastructure to primarily serve multimedia content.



Figure 5.3 Hourly traffic volume timeseries of five adult websites.

5.3.1.3 Temporal Access Patterns

We next analyze the temporal access patterns for adult websites. Figure 5.3 plots the normalized hourly timeseries of traffic volume across the day. We converted the timestamps to local timezones to calculate hourly traffic volumes. Overall, we observe that access patterns of adult websites are *not* typical diurnal patterns. Prior literature (e.g., [145, 112, 126]) reported content access peaks during 7-11 pm and troughs in late night and early morning hours. In contrast, for example, we note that V-1 traffic volume peaks at late-night and early morning hours. This pattern for V-1 is almost opposite to typical diurnal hours reported in prior literature. The temporal access patterns for V-2, P-1, P-2, and S-1 have less pronounced variations than V-1, yet they are different from typical diurnal patterns. The differences between peak access of adult and other contents are likely due to the unique nature and viewing preferences for adult content. Thus, it is important for CDNs to separately account for adult traffic in the traffic forecasting models and network resource allocation.

5.3.1.4 Device/OS Usage

The web traffic is known to be gradually shifting from traditional desktop to smartphones and tablets over the last several years [95]. Recall that we extract user agent information from HTTP headers to identify device/OS of a user. We next investigate the device/OS



Figure 5.4 Device type composition

composition of user requests to adult websites. All adult websites discussed in this chapter have mobile friendly websites/apps. Figure 5.4 plots the device distribution of users accessing the adult websites. We observe that the desktop category dominates smartphones (Android and iOS) and miscellaneous (tablets and other mobile devices) categories. For instance, V-2 has more than 95% users accessing content from traditional desktop devices. We observe that image-heavy and social networking websites receive relatively more visitors from smartphone devices as compared to video websites. For instance, more than one-third of users access S-1 from smartphone and miscellaneous device categories. The differences across different adult websites can be partially explained by user preference to view adult content on larger screens. CDNs can customize content delivery (e.g., compression, encoding) depending on different video playback devices. Since a vast majority of users do not visit adult websites on smartphones, our findings highlight the need for adult websites to further improve their web interfaces and content delivery strategies for mobile devices.

5.3.2 Content Dynamics

5.3.2.1 Content Size

We next investigate content sizes for adult websites. Figure 5.5 plots the Cumulative Distribution Functions (CDFs) of content sizes. We note bi-modal distributions for image objects.



Figure 5.5 Content size distributions.

Small images are low-resolution thumbnails and large images are high-resolution pictures. Overall, we observe that content sizes vary in the range of a few kilobytes (KB) to hundreds of megabytes (MB), where majority of requested video objects have sizes greater than 1 MB and image objects are less than 1 MB in size. Figure 5.5(a) shows the content size distribution of video objects. Video adult websites V-1 and V-2 have a majority of objects larger than 1 MB. P-1 and S-1 have relatively small number of video objects. P-2 has the largest video object sizes. Figure 5.5(b) shows the content size distribution of image objects. We note that multiple adult websites have bi-modal distributions, indicating thumbnail sized images as well as large images of sizes up to 1 MB. These observations have significant implications for CDN/ISP caching optimization. For instance, using separate caching platforms to optimally serve small and large sized objects. The caching platform for small objects can be optimized for high-throughput I/O; and, the caching platform for large objects can be optimized for more storage capacity.

5.3.2.2 Content Popularity

We now investigate object popularity for adult websites. We quantify object popularity in terms of request count. Understanding the popularity of objects is important because CDNs typically optimize their caching performance by focusing on popular objects and



Figure 5.6 Content popularity distributions

reduce storage costs by ignoring unpopular and dynamically changing content. Additionally, analyzing the popularity of adult objects can provide insights for identifying similarities and differences between adult objects and non-adult objects. We plot the request count distribution of video and image objects for adult websites in Figure 5.6. We observe longtail distributions for all adult websites. This observation indicates that a significant fraction of adult objects are requested infrequently and a small fraction of adult objects are very popular. From a content delivery perspective, this information is useful as CDNs can improve their caching performance by caching heavily requested objects. The long-tailed distribution is similar to those reported for traditional web and video content in prior literature, where a smaller fraction of viral media content is heavily requested. The long-tailed distribution also potentially highlights some social aspects of adult content. In comparison, the large number of requests for typical non-adult objects is mainly because of word-of-mouth content sharing in online social networking websites [119]. We would not typically expect adult content viewers to share adult content on popular social networking websites, though recent work has shown that users use social features in adult websites [136]. We next investigate unique aspects of adult content popularity by further analyzing individual object request patterns.



Figure 5.7 Fraction of total object requested for adult websites at different ages



Figure 5.8 Content clustering dendrograms of two adult websites.

5.3.2.3 Impact of Content Injection and Aging on Popularity

We would expect more requests for objects when they are new, and as the content ages we expect its popularity to decrease accordingly. To understand this phenomenon for adult websites, we plot the fraction of adult objects requested at different ages in Figure 5.7. The plot shows that a declining fraction of objects are requested as their age increases. In particular, about 20% of objects are not requested after 3 days for most adult websites. Only about 10% of objects are requested throughout the trace duration of one week.

We further explore the popularity trends of objects by clustering them with respect to their temporal popularity patterns. To identify temporal popularity patterns, we analyze the time series of request count for individual objects. Characterizing the request count



Figure 5.9 Cluster medoids for V-2 adult website



Figure 5.10 Cluster medoids for P-2 adult website

time series helps us to assess how fast an adult object reaches its maximum popularity, identifying governing popularity trends of image and video objects, and developing insights for improving the caching performance of CDNs.

We identify distinct popularity trends by measuring the similarities in shape between the normalized request count time series. We use Dynamic Time Warping (DTW) to compute similarity between two request count time series [102]. DTW uses a dynamic programming approach to obtain a minimum distance alignment between two time series. More specifically, for two time series T_1 and T_2 , DTW obtains an optimal alignment between T_1 and T_2 by warping the time dimension of T_1 and T_2 . DTW achieves an optimal alignment between two time series to points of the second time series. This non-linear mapping is also known as the warping path.

More specifically, for $T_1 = (a_1, a_2, ..., a_N)$ and $T_2 = (b_1, b_2, ..., b_M)$ of length N and M respectively, a warping path $w = (w_1, w_2, ..., w_L)$ defines an alignment between T_1 and T_2 , where $w_1 = (a_1, b_1)$ defines the mapping of element $a_1 \in T_1$ to $b_1 \in T_2$. A cost function is used to compute the optimality of a warping path. Large cost values indicate low similarity in shapes of two time series and small cost values indicate high similarity. The total cost of the a warping path w is defined as:

$$C_w = \sum_{i=1}^{L} c(a_i, b_i)$$

Intuitively, the cost function c is defined as the area between the time warped time series. Using a dynamic programming approach, DTW computes all possible sets of mappings (warping paths) between two time series. The optimal warping path is the path w' that has the minimum total cost among all possible warping paths. The total cost of the optimal warping path is defined as the DTW distance. We use the DTW distance as a metric for quantifying the similarity between two request count time series. We compute pairwise DTW distances for all request count time series. We then use the pairwise DTW distance matrix to obtain hierarchical clusters for the request count time series. We use agglomerative hierarchical clustering to obtain dendrogram for each adult website.

Figures 5.8(a) and (b) show two example dendrograms of video and image objects for V-2 and P-2, respectively. The x-axis of a dendrogram shows the cluster labels and their memberships and the y-axis represents the DTW similarity metric. We observe four dominant popularity trends in our clustering analysis: diurnal, long-lived, and short-lived. We also observe that some objects in V-2 and P-2 websites have other popularity trends that cannot be neatly categorized as any of the aforementioned categories. We categorize these objects as outliers.

To visualize the unique popularity trends, we next identify a representative sample object from each cluster and plot its normalized request count time series with point-wise standard deviations. To identify the representative sample object for each cluster, we identify its *medoid*, where a medoid is defined as the most centrally located point of a cluster [85]. We calculate point-wise standard deviations by looking at the normalized request count timeseries of all objects in the cluster. We plot the normalized request count time series of mediods of unique clusters in Figure 5.9. The shaded regions represent the standard deviation of all timeseries from the mean of the cluster and the solid line represent the medoid of the cluster. Figures 5.9(a) and 5.10(a) show the medoids for the diurnal popularity cluster. Diurnal popularity trends of video objects highlight that certain video objects are requested continuously with regular day/night time variations. Figures 5.9(b) and 5.10(b) show the cluster medoids of objects with long-lived popularity trend. These objects request count reaches its maximum popularity within the first day of their injection. Their request count decays in a diurnal fashion and completely dies down after a few days. Figures 5.9(c)and 5.10(c) show the cluster medoids of objects with short-lived popularity trend. The request count of these objects reaches its maximum during the first day of its arrival but decays sharply and completely dies down within a few hours. Our further analysis reveals that video objects with diurnal trends are smaller in size as compared video objects with long-lived and short-lived trends. The long-lived video objects have the largest size followed by short-lived video objects.

Overall, our analysis reveals that objects served by adult websites have diverse popularity trends. Each website has a different composition of these popularity trends. A large fraction of image and video objects have diurnal request patterns. There are two possible explanations for diurnal patterns. First, prior work [135] suggests that users discover content on adult websites through front page browsing. Objects with diurnal patterns are most likely objects displayed on the front page of a website, and are always delivered when users visit the website. Second, we note that the potentially addictive nature of adult content could drive users to access specific content repeatedly, similar to non-adult media content.

CDNs can utilize this information to optimize cache control by re-validating diurnal objects less frequently and other objects more frequently, for example, hourly for objects with short-lived access patterns and daily for objects with long-lived access patterns. This



Figure 5.11 User request inter-arrival time distributions

can also be achieved by setting longer expire times for objects with diurnal and long-lived access patterns. Furthermore, CDNs can reduce network traffic by pushing copies of objects with diurnal and long-lived request patterns to locations closer to end-users.

5.3.3 User Dynamics

5.3.3.1 User Request Inter-arrival Time

We characterize the user request arrival process in terms of its Inter-arrival time (IAT) distribution. Figure 5.11 plots the user request IAT distributions for all adult websites. Comparing different adult websites, we observe that video adult websites have shorter request IATs as compared to image-heavy adult websites. For video objects in adult websites, the median request IAT is less than 10 minutes, whereas it is more that 1 hour for image-heavy adult websites. We later use these observations for estimating user session lengths.

5.3.3.2 User Session Length

A key metric from the perspective of content publishers and CDNs is user engagement [124], which is typically quantified in terms of website bounce time [57]. From the network-side logs, we can estimate user engagement in terms of user session length,¹ where a session

¹It is noteworthy that the user session length is a strictly lower-bound of traditional bounce time because we cannot tell how long a user continues to watch the downloaded content from network-side logs.



Figure 5.12 User session length distributions



Figure 5.13 Repeated access of objects

consists of consecutive user requests within a timeout interval. We set the timeout value for user sessions at 10 minutes based on our earlier analysis of user request IAT distributions. We plot user session length distributions for adult websites in Figure 5.12. We observe that the median session lengths for most adult websites are around one minute. Our findings indicate that user engagement for adult content consists of relatively short-lived sessions as compared to non-adult content. We further verified our observations using the engagement statistics reported by Alexa. We note that average session lengths for popular non-adult websites tend to be much larger than popular adult websites. For example, the average session length for YouTube [147] is approximately two minutes, whereas the average session



Figure 5.14 CDF of repeated content access by users

length or XVIDEOS [146] is less than one minute.

5.3.3.3 User Addiction

To further investigate user engagement, we next analyze user addiction. We analyze repeated content accesses by a user to investigate content addiction. For each object, we compute the total number of requests and the total number of unique users who make these requests. Figure 5.13(a) shows the scatter plot highlighting repeated access of video objects for V-1 adult website. Each data point in the plot represents a distinct video object. We observe that certain video objects are requested by a user multiple times, i.e., data points above the diagonal. In some cases, an object is requested by a large number of times by an individual user. For example, some objects have up to two orders of magnitude more requests than unique users. This observation indicates that some video objects are popular due repeated access by a certain user (i.e., addiction), whereas other video objects are popular due to multiple users accessing the content (i.e., viral). Figure 5.13(b) shows the scatter plot of repeated access of objects for P-1 adult website. A comparison of repeated access reveal that video objects are more likely to get repeated user requests than image objects. This also highlights that video content is more addictive/engaging as compared to image content.

To further analyze user addiction to content, we plot the distribution of repeated user



Figure 5.15 Hit ratio for image and video objects

access for all objects. Figure 5.14 plots the CDF of number of requests per user for all adult websites. We note that less than 1% of image objects are requested more than 10 times by a user, whereas at least 10% of video objects have more than 10 requests per unique user. From a CDN's perspective, this information is particularly useful as they can differentiate between objects that are popular only due to requests from multiple users versus those objects that are popular because of repeated accesses by a user. This information is also useful in optimizing local browser caching and proxy caches deployed by ISPs. Objects accessed multiple times by a group of users should be locally cached closer to end-users.

5.4 Implications

We discuss potential implications of our measurement and analysis of online adult traffic. We are particularly interested in understanding the impact of different content access patterns on CDN caching. To this end, we analyze the caching performance for adult websites by looking at server-side HTTP response codes and cache hit ratios.



Figure 5.16 Response codes

5.4.1 CDN Cache Hit Ratios

We first investigate the cache performance of CDN servers by analyzing server-side cache hit ratios. Figure 5.15 plots the distributions of cache hit ratios for objects in all adult websites. A hit ratio value of 0 indicates that the requested objects were not present in the CDN cache. A hit ratio value of 1 indicates that all requested objects were served from the CDN cache. Note that the CDN treats video chunks as separate objects for the sake of caching. Comparing image and video objects, we observe that image objects have better overall cache hit ratio than video objects. S-1 has the smallest percentage of objects added to the CDN cache. To understand the cache performance of objects with different popularity, we compute correlations between hit ratio and object popularity. As expected, we find that popular objects tend to have higher hit ratios (more than 0.9 correlation coefficient for all adult websites). Thus, while the distributions in Figure 5.15 may indicate otherwise, overall CDN cache hit ratios range between 80-90% for adult websites. CDNs often customize cache configuration and performance for individual publishers. Thus, differences in cache hit ratios may reflect cache prioritization for different content publishers. Furthermore, customized caching strategies for streaming video content can also be implemented by the CDN.

5.4.2 HTTP Response Codes

We next analyze HTTP response codes for adult websites. Figure 5.16 shows the number of requests associated with each type of response code. The most common HTTP response codes for both video and image objects include: 200, 206, 304, and 403. We note that a majority of response code are 200. Of particular interest to a CDN operator is the 304 response code, which indicates that the client's requested object is not modified and the local cache copy is up to date. We note that 304 responses constitute a small fraction of all requests, which indicates the potential for improved localized content caching to improve user performance and reduce traffic load on CDN content replica servers. Despite the cacheability of popular adult objects, 304 response counts are particularly low for adult websites because users are known to browse adult content in incognito/private browsing modes [16]. Web browsers dispose local cache content when users close the incognito/private browser windows. In contrast, note that Facebook reported that more than 65% of their photo requests are served from local browser caches [74]. Unfortunately, while traditional non-adult website can fully utilize browser cache to improve performance and reduce traffic, adult content publishers cannot solely rely on it in designing their content and caching mechanisms.

5.5 Conclusion

In this chapter, we presented a large scale and in-depth measurement and analysis of online adult traffic. We provide an in-depth analysis of their aggregate, content, and user dynamics. We find that the temporal access patterns of adult website are unique and different from typical diurnal access patterns. While a majority of users access adult websites from desktop, smartphones and other mobile devices account for a non-trivial fraction of visitors. Our clustering analysis of content popularity reveals groups of objects with diurnal, long-lived, and short-lived temporal access patterns. User engagement in adult websites is shorter than non-adult websites; however, adult content can be addictive, i.e., some users repeatedly access certain content. Our findings have implications on adult website design and content delivery infrastructure management. For instance, adult content providers cannot rely on browser cache to store locally popular content because of the prevalent usage of incognito/private web browsing. Content delivery networks can also reduce improve performance and network traffic by pushing copies of popular adult objects to locations closer to the end-users.

6 Privacy Preserving Social Network Graph Data Publishing

6.1 Introduction

6.1.1 Background and Motivation

Online Social Networks (OSNs) have become an essential part of modern life. Billions of users connect and share information using OSNs such as Facebook and Twitter. Graphs obtained from these OSNs can provide useful insights on various fundamental societal phenomena such as epidemiology, information dissemination, marketing, and sentiment flow [58, 118]. Various analysis methods [38, 92] have been applied to OSNs by explicitly exploring its graph structure, such as clustering analysis for automatically identifying online communities and node influence analysis for recognizing the influential nodes in social networks. The basis of all these analysis is to represent a social network graph by an adjacency matrix and then represent individual nodes by vectors derived from the top eigenvectors of the adjacency matrix. Thus, all these analysis methods require real social network graphs.

Unfortunately, OSN companies often refuse to publish their social network graphs due to privacy concerns. Social network graphs contain sensitive information about individuals such as an user's topological characteristics in a graph (e.g., number of social ties, influence in a community, etc). From the user perspective, the sensitive information revealed from a social network graph can be exploited in many ways such as fraud and spam. From the OSN perspective, disclosing sensitive user information may put them in the risk of violating privacy laws. A natural way to bridge the gap is to anonymize original social network graphs (by means such as removing identifiers) and publish the anonymized ones. For example, Netflix published anonymized movie ratings of 500,000 subscribers and AOL published search queries of 658,000 users [63, 105]. However, such anonymization is vulnerable to privacy attacks where attackers can identify personal information by linking two or more databases [31].

6.1.2 Problem Statement

In this work, we aim to develop a scheme for publishing social network graphs with differential privacy guarantees. Recently, differential privacy has become the widely accepted criteria for privacy preserving data publishing because it provides strongest privacy guarantees for publishing sensitive datasets [47]. The concept of differential privacy was proposed by Dwork in the context of statistical databases, where a trusted party holds a dataset D containing sensitive information (e.g. medical records) and wants to publish a dataset D' that provides the same global statistical information as D while preserving the private information of each individual user. Given a database D and its neighbor database D' differing from D in only one record, according to differential privacy, any information that can be learned from the database D can also be learned from D', irrespective of the presence or absence of a record. Our targeted differential privacy preserving social graph publishing scheme should satisfy the following two requirements. First, the published dataset should maintain the utility of the original dataset. Second, the scheme should guarantee differential privacy that is, an adversary should learn nothing more about any individual from the published data, regardless of the presence or absence of the record of any particular individual in the dataset. We emphasize that these two goals are often conflicting: to achieve differential privacy, a sufficiently large amount of random noise has to be added to the published data, which could potentially result in a large error in approximating the top eigenvectors of the original data. The goal of this work is to achieve a proper tradeoff between privacy and utility.

6.1.3 Limitations of Prior Art

Given an original $n \times n$ matrix, some schemes have been proposed to perturb this matrix by adding random noise and then publish the perturbed $n \times n$ matrix [40, 82]. The key limitation of such schemes is that they are computationally unpractical as they are not designed to handle large matrices such as the adjacency matrices of OSN graphs, which can be in the scale of millions or even billions. Recently, Wang *et al.* proposed a scheme called LNPP, which perturbs the eigenvectors of the original matrices by adding random noise and then publishes the perturbed eigenvectors [141]. Although this reduces computation cost and storage space, this scheme requires a large amount of random perturbation in order to preserve differential privacy, which leads to poor estimation of the top eigenvectors.

6.1.4 Proposed Approach

We propose a random matrix approach by leveraging the theories of random matrix and differential privacy. Our key idea is to first project each row of an adjacency matrix into a low dimensional space using random projection, and then perturb the projected matrix with random noise, and finally publish the perturbed and projected matrix. Given a $n \times n$ adjacency matrix A of a social network graph, with m projections and k eigenvectors, we output an $n \times m$ ($0 < k \leq m \ll n$) randomly perturbed matrix \widehat{A} where the top k eigenvectors of A are close to the k eigenvectors of \widehat{A} (in terms of both vector directions and magnitude). The random projection is critical in our approach. First, it significantly reduces the size of the matrix to be published because $n \times m \ll n \times n$, addressing the limitations of prior art. Second, based on the theory of random matrix [61], we can preserve the top eigenvectors of the adjacency matrix using random projection. Third, the random projection step by itself has the ability of achieving differential privacy, which makes it possible to ensure differential privacy in the second step by introducing a *small* random perturbation [113, 34].

6.1.5 Technical Challenges

There are three key technical challenges. The first challenge is to prove that random projection plus random perturbation preserve differential privacy. The second challenge is to prove that the random noise required to achieve differential privacy is small. We address these two challenges by leveraging the theory of random matrix and differential privacy. The third challenge is to validate the proposed approach and demonstrate the utility preservation of eigen-spectrum. We address this challenge by evaluating the utility of the published data for two different applications that require the spectral information of an OSN graph. We use publicly available datasets of Facebook, Live Journal and Pokec social networks [11, 143]. The first application is OSN node clustering, which has been widely used for OSN applications such as community detection [139]. We focus on spectral clustering algorithms as they are based on the eigenvectors of adjacency matrices. The second application is OSN node ranking, which has been widely used for OSN applications such as influential node identification [75]. We focus on principal component centrality based algorithms as they are based on the eigenvectors of adjacency matrices.

6.1.6 Key Contributions

We make three key contributions in this chapter. First, we propose a random matrix approach to OSN graph publishing, which achieves space efficiency by reducing the dimensions of adjacency matrices and achieves differential privacy by adding a small amount of noise. Second, we formally prove that our scheme achieves differential privacy and that the amount of added noise is small. We present theoretical error bounds for approximating top-k eigenvectors. Third, we validate our random matrix approach on two different applications: node clustering, and node ranking, which require the spectral information of a graph. Our experimental results show that even for high values of noise variance $\sigma = 1$, our scheme preserves the utility of the dataset. For node clustering, our scheme achieves high clustering quality defined by normalized mutual information, which is more than 0.7. For node ranking, our scheme can correctly recover 80% of the most influential nodes. For node classification with 16 classes, our scheme obtains an accuracy of 70% with 5-fold cross validation. We also compare our results with the LNPP scheme, which represents the state-of-the-art [141]. We show that our approach achieve better results for both clustering and ranking. The LNPP scheme directly perturbs the eigenvector of the original data by a Laplacian noise; in contrast, we first use random projections to reduce the size of the adjacency matrix and then compute the eigenvectors. Our approach achieves a high clustering quality of 0.74, which is the normalized mutual information; for LNPP, the same amount of noise almost completely destroys the clustering of nodes in the original dataset. For node ranking, our approach correctly identifies 80% of the most influential nodes, whereas LNPP correctly identifies less than 1% of the most influential nodes.

6.2 Related Work

In the area of differential privacy in data publishing, some schemes have been proposed to perturb an $n \times n$ matrix by adding random noise and then publish the perturbed $n \times n$ matrix [40, 82]. Given a large $n \times n$ matrix, no matter dense or spare, all these schemes output another large dense matrix of size $n \times n$. The key limitation of such schemes is that they are computationally unpractical as in large OSNs, n is in the scale of millions or even billions. Blum *et al.* propose to publish the covariance matrix of the original data contaminated by random noise [35]. [113] and [34], show that random projection by itself can preserve both the differential privacy and the eigen-spectrum of a given matrix provided appropriate modification is made to the original matrix. Hardt *et al.* proposed an iterative algorithm to compute differential private eigenvectors [64]. This scheme generates large dense matrix of $n \times n$ at each iteration. Sampling approaches based on the exponential mechanism have been proposed for computing differential private singular vectors [40, 82]. Since these approaches require sampling very high dimensional vectors from a random distribution, they are computationally infeasible for large social networks. Wang *et al.* propose to publish eigenvectors perturbed by Laplacian random noise, which unfortunately requires a large amount of random perturbation for differential privacy preservation and consequentially leads to poor utility of data [141].

Some other work has been done on publishing social network data with differential privacy; however, none of them addresses the utility of preserving eigenvectors of social network graphs, which is the main goal of our work [120, 66, 101, 86]. Sala *et al.* presented a scheme to share meaningful graph datasets, based on the dk-graph model, while preserving differential privacy [120]. Hay *et al.* proposed a scheme to preserve the degree distribution of a social network graph [66]. Mir and Wright proposed to guarantee differential privacy by perturbing Kronecker model parameters [101]. Kenthapadi developed a differential private algorithm that preserves distance between any two samples in a given database [86]. Although these studies deal with differential private publication of social network data, none of them address the utility of preserving the eigenvectors of the graph, the central theme of this work.

6.3 Random Matrix Approach

In this section, we first present the proposed approach for differential private publication of social network graph based on the random matrix theory. We then present its guarantee on differential privacy and the approximation of eigenvectors.

Let G be a binary graph representing the connectivity of a social network, and $A \in \{0,1\}^{n \times n}$ be the adjacency matrix representing the graph, where $A_{i,j} = 1$ if there is an edge between nodes i and j, and $A_{i,j} = 0$, otherwise. The first step of our approach is to generate two Gaussian random matrices $P \in \mathbb{R}^{n \times m}$ and $Q \in \mathbb{R}^{m \times m}$, where $m \ll n$ is the number of random projections. Here, each entry of P is sampled independently from a Gaussian distribution $\mathcal{N}(0, 1/m)$, and each entry of σ will be discussed later. Using Gaussian

random matrix P, we compute the projection matrix $A_p \in \mathbb{R}^{n \times m}$ by $A_p = A \times P$, which projects each row of A from a high dimensional space \mathbb{R}^n to into a low dimensional space \mathbb{R}^m . We then perturb A_p with the Gaussian random matrix Q by $\widehat{A} = A_p + Q$, and publish \widehat{A} to the external world. Algorithm 3 highlights the key steps of the proposed routine for publishing the social network graph.

Algorithm 3 $\widehat{A} = \text{Publish}(A, m, \sigma^2)$
Input: (1) symmetric adjacency matrix $A \in \mathbb{R}^{n \times n}$
(2) the number of random projections $m < n$
(3) variance for random noise σ^2
Output: \hat{A}

- 1 Compute a random projection matrix P, with $P_{i,j} \sim \mathcal{N}(0, 1/m)$
- **2** Compute a random perturbation matrix Q, with $Q_{i,j} \sim \mathcal{N}(0, \sigma^2)$
- **3** Compute the projected matrix $A_p = AP$
- 4 Compute the randomly perturbed matrix $\hat{A} = A_p + Q$

Compared to the existing approaches for differential private publication of social network graphs, the proposed algorithm is advantageous in three aspects: First, the proposed algorithm is computationally efficient as it does not require either storing or manipulating a *dense* matrix of $n \times n$. Second, the random projection matrix P allows us to preserve the top eigenvectors of A due to the theory of random matrix. Third, the joint effort of the random projection P and the random perturbation Q preserves differential privacy. This unique feature allows us to introduce a small amount of random perturbation for differential privacy preservation, thus improving the utility of data.

Next, we give a theoretical analysis of two main aspects of publishing differential private graphs of OSNs. First, we formally prove that our random matrix approach to publishing OSN graphs guarantees differential privacy. Second, we give theoretical error bounds for approximating top-k eigenvectors.

6.3.1 Theoretical Guarantee on Differential Privacy

Before we show the guarantee on differential privacy, we first introduce the definition of (ϵ, δ) -Differential Privacy: A (randomized) algorithm \mathcal{A} satisfies (ϵ, δ) -differential privacy, if for all inputs X and X_0 differing in at most one user's one attribute value, and for all sets of possible outputs $D \subseteq Range(\mathcal{A})$, we have

$$\Pr\left(\mathcal{A}(X) \in D\right) \le e^{\epsilon} \Pr\left(\mathcal{A}(X_0) \in D\right) + \delta,\tag{6.1}$$

where the probability is computed over the random coin tosses of the algorithm.

To understand the implication of (ϵ, δ) -differential privacy, consider the database $X \in \{0, 1\}^{n \times m}$ as a binary matrix. Let $p_{i,j} := \Pr(X_{i,j} = 1)$ represent the prior knowledge of an attacker about X, and let $p'_{i,j} = \Pr(X_{i,j} = 1 | \mathcal{A}(X))$ represent his knowledge about Xafter observing the output $\mathcal{A}(X)$ from algorithm \mathcal{A} . Then, if an algorithm \mathcal{A} satisfies (ϵ, δ) differential privacy, then with a probability $1 - \delta$, for any $i \in [n]$ and $j \in [m]$, the following condition holds:

$$\left|\ln p_{i,j} - \ln p'_{i,j}\right| \le \epsilon$$

In other words, the additional information gained by observing $\mathcal{A}(X)$ is bounded by ϵ . Thus, parameter $\epsilon > 0$ determines the degree of differential privacy: the smaller the ϵ , the less the amount of information will be revealed. Parameter $\delta \in (0, 1)$ is introduced to account the rare events when the two probabilities $\Pr(\mathcal{A}(X) \in D)$ and $\Pr(\mathcal{A}(X_0) \in D)$ may differ significantly from each other.

Theorem 1. Assuming $\delta < 1/2$, $n \ge 2$, and

$$\sigma \geq \frac{1}{\epsilon} \sqrt{10\left(\epsilon + \ln\frac{1}{2\delta}\right) \ln\frac{n}{\delta}}$$

Then, Algorithm 3 satisfies (ϵ, δ) -differential privacy w.r.t. a change in an individual person's attribute.

The key feature of Theorem 1 is that the variance for generating the random perturbation matrix Q is $O(\ln n)$, independent from the size of social network. As a result, we can ensure

differential privacy for the published \widehat{A} for a very large social network by only introducing a Gaussian noise with small variance, an important feature that allows us to simultaneously preserve both the utility and differential privacy. Our definition of differential privacy is a generalized version of ϵ -differential privacy which can be viewed as (ϵ , 0)-differential privacy.

6.3.2 Proof of Theorem 1

To prove that Algorithm 3 is differential private, we need the following Theorem from [86]

Lemma 6.3.1. (Theorem 1 [86]) Define the ℓ_2 -sensitivity of the projection matrix P as $w_2(P) = \max_{1 \le i \le n} |P_{i,*}|_2$, where $P_{i,*}$ represents the *i*th row of matrix P. Assuming $\delta < 1/2$, and

$$\sigma \ge \frac{w_2(P)}{\epsilon} \sqrt{2\left(\epsilon + \ln \frac{1}{2\delta}\right)}$$

Then Algorithm 3 satisfies (ϵ, δ) -differential privacy w.r.t. a change in an individual person's attribute.

In order to bound $w_2(P)$, we rely on the following concentration for χ^2 distribution.

Lemma 6.3.2. (Tail bounds for the χ^2 distribution) Let X_1, \ldots, X_d be independent draws from $\mathcal{N}(0,1)$. Therefore, for any $0 < \delta < 1$, we have, with a probability $1 - \delta$,

$$\sum_{i=1}^{d} X_i^2 \le d + 2\sqrt{d\ln\frac{1}{\delta}} + 2\ln\frac{1}{\delta}$$

Define

$$z_i^2 = \sum_{j=1}^m P_{i,j}^2$$

Evidently, according to the definition of $w_2^2(P)$, we have

$$w_2^2(P) = \max_{1 \le i \le n} z_i^2$$

Since $P_{i,j} \sim \mathcal{N}(0, 1/m)$, we have mz_i^2 follow the χ^2 distribution of d freedom. Using Lemma 3.2, we have, with a probability $1 - \delta$,

$$z_i^2 \le 1 + 2\sqrt{\frac{1}{m}\ln\frac{1}{\delta}} + \frac{2}{m}\ln\frac{1}{\delta} \tag{6.2}$$

By taking the union bound, we have, with a probability $1 - \delta$

$$w_2^2(P) = \max_{1 \le i \le m} z_i^2 \le 1 + 2\sqrt{\frac{1}{m} \ln \frac{n}{\delta}} + \frac{2}{m} \ln \frac{n}{\delta} \le 2$$
(6.3)

where the last inequality follows from $m \ge 4 \ln(n/\delta)$. We complete the proof by combining the result from Lemma 3.1 and inequality in (6.3). Inequality (6.3) comes from the inequality (6.2). According to inequality (6.2), each z_i can fail inequality (6.2) with a probability at most δ . Then, for all z_i , to ensure that the chance for inequality (6.2) to fail with probability at most δ , we can simply replace δ in (6.2) with δ/n , which immediately leads to the inequality (6.3). Therefore, Theorem 1 holds because of Lemma 1 and inequality (6.3).

6.3.3 Theoretical Guarantee on Eigenvector Approximation

Let u_1, \ldots, u_n the eigenvectors of the adjacency matrix A ranked in the descending order of eigenvalues $\lambda_1, \ldots, \lambda_n$. Let k be the number of top eigenvectors of interests. Let $\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_k$ be the first k eigenvectors of \hat{A} . Define the approximation error for the first k eigenvectors as

$$\mathcal{E}^2 = \max_{1 \le i \le k} |\mathbf{u}_i - \widetilde{\mathbf{u}}_i|^2$$

Our goal is to show that the approximation error \mathcal{E}^2 will be small when the number of random projections *m* is sufficiently large.

Theorem 2. Assume (i) $m \ge c(k + k \ln k)$, where c is an universal constant given in [121], (ii) $n \ge 4(m + 1) \ln(12m)$ and (iii) $\lambda_k - \lambda_{k+1} \ge 2\sigma\sqrt{2n}$. Then, with a probability at least 1/2, we have

$$\mathcal{E}^2 \le \frac{16\sigma^2 n}{(\lambda_k - \lambda_{k+1})^2} + \frac{32}{\lambda_k^2} \sum_{i=k+1}^n \lambda_i^2$$

The corollary below simplifies the result in Theorem 2 by assuming that λ_k is significantly larger than the eigenvalues $\lambda_{k+1}, \ldots, \lambda_n$. **Corollary 6.3.1.** Assume (i) $\lambda_k = \Theta(n/k)$, and (ii) $\sum_{i=k+1}^n \lambda_i^2 = O(n)$. Under the same assumption for m and n as Theorem 2, we have, with a probability at least 1/2,

$$\mathcal{E} \le O\left(k\left[\frac{\sigma}{\sqrt{n}} + \frac{1}{\sqrt{n}}\right]\right)$$

As indicated by Theorem 2 and Corollary 6.3.1, under the assumptions (i) λ_k is significantly larger than eigenvalues $\lambda_{k+1}, \ldots, \lambda_n$, (ii) the number of random projections m is sufficiently larger than k, and (iii) n is significantly larger than the number of random projections m, we will have the approximation error $\mathcal{E} \propto O(k/\sqrt{n})$ in recovering the eigenvectors of the adjacency matrix A. We also note that according to Corollary 6.3.1, the approximation error is proportional to σ , which measures the amount of random perturbation needed for differential privacy preservation. This is consistent with our intuition, *i.e.*, the smaller the random perturbation, the more accurate the approximation of eigenvectors.

6.3.4 Proof of Theorem 2

Next, we prove Theorem 2. Let $A \in \mathbb{R}^{n \times n}$ be the adjacency matrix, $A_p = AP$, and $\widehat{A} = A_p + Q$. Let $\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_k$ be the first k eigenvectors of matrix A_p . Define $U = (\mathbf{u}_1, \ldots, \mathbf{u}_k)$, $\widehat{U} = (\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_k)$, and $\widetilde{U} = (\widetilde{\mathbf{u}}_1, \ldots, \widetilde{\mathbf{u}}_k)$. For each of these matrices, we define a projection operator, denoted by P_k , \widehat{P}_k and \widetilde{P}_k , as

$$P_{k} = \sum_{i=1}^{k} \mathbf{u}_{i} \mathbf{u}_{i}^{\top} = UU^{\top}$$
$$\widehat{P}_{k} = \sum_{i=1}^{k} \widehat{\mathbf{u}}_{i} \widehat{\mathbf{u}}_{i}^{\top} = \widehat{U}\widehat{U}^{\top}$$
$$\widetilde{P}_{k} = \sum_{i=1}^{k} \widetilde{\mathbf{u}}_{i} \widetilde{\mathbf{u}}_{i}^{\top} = \widetilde{U}\widetilde{U}^{\top}$$

We first bound the approximation error \mathcal{E}^2 by the difference between projection operators, *i.e.*,

$$\mathcal{E}^2 = \max_{1 \le i \le k} |\mathbf{u}_i - \widetilde{\mathbf{u}}_i|^2 \le \|UU^\top - \widetilde{U}\widetilde{U}^\top\|_2 = \|P_k - \widetilde{P}_k\|_2$$

where $\|\cdot\|_2$ stands for the spectral norm of matrix. Using the fact that

$$\mathcal{E}^{2} \leq \|P_{k} - \widetilde{P}_{k}\|_{2}^{2} = \|P_{k} - \widehat{P}_{k} + \widehat{P}_{k} - \widetilde{P}_{k}\|_{2}^{2} \\
\leq 2\|P_{k} - \widehat{P}_{k}\|_{2}^{2} + 2\|P_{k} - \widetilde{P}_{k}\|_{2}^{2} \\
\leq 2\|P_{k} - \widehat{P}_{k}\|_{F}^{2} + 2\|P_{k} - \widetilde{P}_{k}\|_{2}^{2}$$
(6.4)

where $\|\cdot\|_F$ stands for the Frobenius norm of matrix, below we will bound $\|P_k - \hat{P}_k\|_F$ and $\|P_k - \tilde{P}_k\|_F$, separately.

To bound $||P_k - \hat{P}_k||_F$, we need the following theorem for random matrix.

Lemma 6.3.3. (Theorem 14 [121]) Assume $0 < \epsilon \le 1$ and $m \ge c(k/\epsilon + k \ln k)$, where c is some universal constant. Then, with a probability at least 2/3, we have

$$||A - \widehat{P}_k(A)||_F \le (1 + \epsilon) ||A - P_k(A)||_F,$$

Since

$$\begin{split} \|A - \widehat{P}_{k}(A)\|_{F} &\geq -\|A - P_{k}(A)\|_{F} + \|P_{k}(A) - \widehat{P}_{k}(A)\|_{F} \\ &= -\|A - P_{k}(A)\|_{F} - \|P_{k}(A) + \widehat{P}_{k}P_{k}(A) + \widehat{P}_{k}P_{k}(A) - \widehat{P}_{k}(A)\|_{F} \\ &\geq -\|A - P_{k}(A)\|_{F} + \|P_{k}(A) - \widehat{P}_{k}P_{k}(A)\|_{F} - |\widehat{P}_{k}(A - P_{k}(A))\|_{F} \\ &\geq \|P_{k}(A) - \widehat{P}_{k}P_{k}(A)\| - 2\|A - P_{k}(A)\|_{F} >, \end{split}$$

combining with the result from Lemma 3, we have, with a probability at least 2/3,

$$\|(P_k - \hat{P}_k P_k)(A)\|_F \le (3 + \epsilon)|A - P_k(A)|_F$$
(6.5)

Since

$$\|(P_{k} - \widehat{P}_{k}P_{k})(A)\|_{F}$$

= $\|(P_{k}P_{k} - \widehat{P}_{k}P_{k})(A)\|_{F} = \|(P_{k} - \widehat{P}_{k})P_{k}(A)\|_{F}$
\geq $\|P_{k} - \widehat{P}_{k}\|_{F}\|P_{k}(A)\|_{2} = \lambda_{k}\|P_{k} - \widehat{P}_{k}\|_{F}$
combining with the inequality in (6.5), we have, with a probability at least 2/3,

$$\|P_k - \widehat{P}_k\|_F \le \frac{3+\epsilon}{\lambda_k} |A - P_k(A)|_F \tag{6.6}$$

In order to bound $\|\widehat{P}_k - \widetilde{P}_k\|_2$, we use the Davis-Kahan $\sin\Theta$ theorem given as below.

Lemma 6.3.4. Let A and \tilde{A} be two symmetric matrices. Let $\{\mathbf{u}_i\}_{i=1}^k$ and $\{\tilde{\mathbf{u}}_i\}_{i=1}^k$ be the first k eigenvectors of A and \tilde{A} , respectively. Let $\lambda_k(A)$ denote the kth eigenvalue of A. Then, we have

$$\begin{split} \|P_k - \widetilde{P}_k\|_2 &\leq \frac{\|A - A\|_2}{\lambda_k(A) - \lambda_{k+1}(\widetilde{A})} \\ \text{if } \lambda_k(A) > \lambda_{k+1}(\widetilde{A}), \text{ where } P_k = \sum_{i=1}^k \mathbf{u}_k \mathbf{u}_k^\top \text{ and } \widetilde{P}_k = \sum_{i=1}^k \widetilde{\mathbf{u}}_i \widetilde{\mathbf{u}}_i^\top \end{split}$$

Using Lemma 4 and the fact

$$\lambda_{k+1}(\widehat{A}) \le \lambda_{k+1}(A_p) + ||A_p - \widehat{A}||_2 = \lambda_k + ||Q||_2$$

we have

$$\begin{aligned} \|\widehat{P}_k - \widetilde{P}_k\|_2 &\leq \frac{\|A_p - A\|_2}{\lambda_k(A_p) - \lambda_{k+1}(\widehat{A})} \\ &\leq \frac{\|Q\|_2}{\lambda_k - \lambda_{k+1} - \|Q\|_2} \end{aligned}$$

Under the assumption that $\lambda_k - \lambda_{k+1} \ge 2 \|Q\|_2$, we have

$$\|\widehat{P}_k - \widetilde{P}_k\|_2 \le \frac{2\|Q\|_2}{\lambda_k - \lambda_{k+1}}$$

In order to bound the spectral norm of Q, we need the following lemma from random matrix.

Lemma 6.3.5. Let $A \in \mathbb{R}^{r \times m}$ be a standard Gaussian random matrix. For any $0 < \epsilon \le 1/2$, with a probability at least $1 - \delta$, we have

$$\left\|\frac{1}{m}AA^{\top} - I\right\|_{2} \le \epsilon$$

provided

$$m \ge \frac{4(r+1)}{\epsilon^2} \ln \frac{2r}{\delta}$$

Using Lemma 5 and the fact that $Q_{i,j} \sim \mathcal{N}(0, \sigma^2)$, we have, with a probability at least 5/6

$$\|QQ^\top\|_2 \le (1+\eta)\sigma^2 n$$

where

$$n \ge \frac{4(m+1)}{\eta^2} \ln(12m)$$

As a result, we have, with a probability at least 5/6,

$$\|Q\|_2 \le \sigma \sqrt{(1+\eta)n}$$

and therefore

$$\|\widehat{P}_k - \widetilde{P}_k\|_2 \le \frac{2\sigma}{\lambda_k - \lambda_{k+1}}\sqrt{(1+\eta)n} \tag{6.7}$$

We complete the proof by combining the bounds for $||P_k - \hat{P}_k||_F$ and $||\hat{P}_k - \tilde{P}_k||_2$ in (6.6) and (6.7) and plugging them into the inequality in (6.4).

6.4 Experimental Results

To demonstrate the effectiveness of our differential private random matrix approach and to illustrate the utility preservation of eigen-spectrum, we perform experiments over graphs obtained from three different online social networks. We analyze the impact of perturbation by evaluating the utility of the published data for two different applications which require spectral information of a graph. First, we consider clustering of social networks, which has been widely used for community detection in social networks. We choose spectral clustering algorithm in our study, which depends on the eigenvectors of the adjacency matrix. Next, we examine the problem of identifying the ranks of influential nodes in a social network graph. For the evaluation purposes, we obtain clusters and node ranks from the published graph, and compare the results against those obtained from the original graph. We give a brief description of the results obtained for each of the applications of graph spectra in the subsequent sections.



Figure 6.1 Degree distribution of three datasets.

6.4.1 Dataset

In our evaluation we use three different social network graphs from Facebook, Live Journal and Pokec. We use the Facebook data set collected by Wilson et al. from Facebook [143]. The social graphs of Live Journal and Pokec were obtained from publicly available SNAP graph library. The choice of these social networks is based on two main requirements. First, the network should be large enough so that it is a true representation of real online social structure. A small network not only under-represents the social structure, but also produces biased results. Second, the number of edges in the network should be sufficiently large in order to reveal the interesting structure of the network. For all three benchmark datasets, the ratio of the number of edges to the number of nodes is between 7 and 20. Table 6.1 provides the basic statistics of the social network graphs. Figure 6.1 shows degree distribution of 3 online social networks on log-log scale. We can see that the data follows a power law distribution which is a characteristic of social network degree distribution.

Network	Nodes	Edges
Facebook	3,097,165	23,667,394
Pokec	1,632,803	30,622,564
LiveJournal	3,997,962	34,681,189

Table 6.1 Dataset description



Figure 6.2 NMI values for Facebook



Figure 6.3 NMI values for Live Journal

6.4.2 Node Clustering

Clustering is a widely used technique for identifying groups of similar instances in a data. Clustering has applications in community detection, targeted marketing, bioinformatics etc. Social networks posses large amount of information which can be utilized in extensive data mining applications. Large complex graphs can be obtained from social networks which represent relationships among individual users. One of the key research questions is the understanding of community structure present in large social network graphs. Social net-



Figure 6.4 NMI values for Pokec

working platforms possess strong community structure of users, which can be captured by clustering nodes of a social network graph. Detecting communities can help in identifying structural position of nodes in a community. Nodes with a central position in a community have influence in the community. Similarly, nodes lying at the intersection of two communities are important for maintaining links between communities. Disclosure of the identity of such nodes having important structural properties results in serious privacy issues. Therefore, in order to protect an individual's privacy it is crucial for data publishers to provide rigorous privacy guarantees for the data to be published. In our experiments, we use spectral clustering for evaluating our privacy-preserving random matrix approach. Spectral clustering has many fundamental advantages over other clustering algorithms. Unlike other clustering algorithms, spectral clustering is particularly suitable for social networks, since it requires an adjacency matrix as an input and not a feature representation of the data. For social network data graph G represented by the binary adjacency matrix A, spectral clustering techniques utilize the eigen-spectrum of A to perform clustering. The basic idea is to view clustering as a graph partition problem, and divide the graph into several disjoint subgraphs by only removing the edges that connect nodes with small similarities.

In order to evaluate the utility of the published data for clustering, we utilize normalize mutual information (NMI) as a measure to evaluate the quality of clustering [59]. Although Purity is a simpler evaluation measure, high purity is easy to achieve for large number of

clusters and cannot be used to evaluate trade off between quality of clustering and number of clusters. NMI allows us to evaluate this tradeoff by normalizing mutual information $I(\omega; C)$ as described in Equation 6.8.

$$NMI = \frac{I(\omega; C)}{[H(\omega) + H(C)]/2},\tag{6.8}$$

where H is entropy which measures the uniformity of the distribution of nodes in a set of clusters, $\omega = w_1, ..., w_k$ is a set of clusters and $C = c_1, ..., c_k$ is a set of classes or ground truth. NMI is bounded between 0 and 1, and the larger the NMI, the better the clustering performance is. We perform extensive experiments over the datasets to evaluate our approach. We now give a stepwise explanation of our evaluation protocol. Since we do not have ground truth about the communities in the datasets, we employ an exhaustive approach to evaluate clustering over the original data and generate the ground truth communities. First, for a given value of k we generate 5 different sets of clusters from spectral clustering algorithm, represented as C_i for i = 1, ..., 5. Since spectral clustering employs k-means, each set C_i can have different cluster distributions. Therefore, to evaluate the consistency in cluster distribution, NMI values are obtained for $\binom{5}{2}$ different pairs of sets represented as (C_i, C_j) , where $i \neq j$ and average value is reported. Then, another 5 cluster sets are obtained through differential private spectral clustering, represented as ω_i for i = 1, ..., 5. Finally, to evaluate cluster sets ω_i , NMI values are obtained using C_i as the ground truth. In this case NMI values are obtained for each pair (ω_i, C_j) $\forall i, j \in 1, ..., 5$ and the average value is reported.

We evaluate the clustering results for three different values of σ , suggested in [82]. For each σ , we evaluate clustering for two different number of random projections m = 20, 200. Figure 6.2, 6.3 and 6.4 shows NMI values obtained for four different values of k, where symbol O represents the NMI values obtained by using the original data. It is not surprising to observe that the clustering quality deteriorates with increasing number of clusters. This is because the larger the number of clusters, the more the challenging the problem is. Overall, we observe that m = 200 yields significantly better clustering performance than m = 20.



Figure 6.5 Cluster distribution for Facebook

When the random perturbation is small (i.e. $\sigma = 0.1$), our approach with m = 200 random projections yields similar clustering performance as spectral clustering using the original data. This is consistent with our theoretical result given in Theorem 2, i.e. with sufficiently large number of random projections, the approximation error in recovering the eigenvectors of the original data can be as small as $O(\sigma/\sqrt{n})$. Note that the NMI values represented by the symbol O, are obtained for pairs of clusters (C_i, C_j) . These NMI values represents the quality of clustering when clusters C_i are compared with clusters C_j . The NMI values represented by m = 20,200, are obtained when clusters C_j are compared with clusters ω_i . Finally, we observe that the clustering performance declines with larger noise for random perturbation. However, even with random noise as large as $\sigma = 1$, the clustering performance using the differential private copy of the social network graph still yield descent performance with NMI ≥ 0.70 . This is again consistent with our theoretical result: the approximation error of eigenvectors is $O(\sigma/\sqrt{n})$, and therefore will be small as long as σ is significantly smaller than \sqrt{n} . Table 6.2 shows that both the memory requirement and running time increases significantly with increasing number of random projections. To show the variation in the cluster distribution, we select clusters obtained from Facebook data for k = 200 and $\sigma = 1$. Figure 6.5, 6.6 and 6.7 shows the percentage of nodes present in clusters obtained from the original and published data. Note that perturbation has little to no effect over small number of clusters as the distribution of nodes is identical. We compare our results with an approach presented by [141], which directly perturbs the eigenvector of the original



Figure 6.6 Cluster distribution for Live Journal



Figure 6.7 Cluster distribution for Pokec

Dataset	Facebook	Pokec	LiveJournal
Memory (MB) $m = 200$	4955	2612	6396
Memory (MB) $m = 20$	495	261	639
Time $(\sec)m = 200$	150	97	211
Time $(\sec)m = 20$	6.15	4.60	8.15

Table 6.2 Memory utilization and running time for the proposed algorithm

data by a Laplacian noise. We refer to this approach as (LNPP) for short. We note that we did not compare to the other approaches for differential private eigen decomposition because they are computationally infeasible for the large social networks studied in our experiments. We implement the LNPP mechanism and evaluate the clustering performance by comparing it to the clustering results generated by the original adjacency matrix. Table 6.3 gives NMI results using LNPP over different datasets for $\sigma = 1$. It is clear that LNPP performs significantly worse than the proposed algorithm in clustering. Note that we did not include the clustering performance of LNPP in Figure 6.2, 6.3 and 6.4 because of its poor performance that basically overlaps with the horizonal axis.

Cluster	2	4	8	16
Facebook	9.1E - 8	8.8E - 7	4.1E - 6	1.3E - 5
LiveJournal	9.7E - 7	3.2E - 6	3.6E - 6	1.1E - 5
Pokec	1.1E - 7	3.5E - 6	5.8E - 6	2.6E - 5

Table 6.3 NMI for clustering using LNPP Approach [141] for $\sigma = 1$



Figure 6.8 Percentage of preserved ranks using random matrix approach

6.4.3 Node Ranking

Identifying information hubs in a social network is an important problem. An information hub refers to a node which occupies a central position in the community and has a large number of connections with other users. Such central nodes play an important role in information diffusion. Advertising agencies, can utilize information about top-t influential nodes for word-of-mouth advertisements [97]. Therefore, the preservation of privacy of such influential nodes is important. Influential node analysis require information about the eigenspectrum of the social network graph. Eigen-vector centrality (EVC) is a measure to quantify the influence of a node in a social network [36]. EVC is mathematically related to several other influence measures such as [84, 68]. EVC requires the computation of eigen-vectors and assigns ranks to nodes according to their location in the most dominant community. EVC of an adjacency matrix is defined as its principle eigenvector. We employ principal component centrality (PCC) which is based on EVC measure to rank the nodes [76]. Let kdenote the number of eigen vectors to be computed. Let U denote the $n \times k$ matrix whose *ith* column represents the *ith* eigenvector of an $n \times n$ adjacency matrix A. Then PCC can be expressed as:

$$C_k = \sqrt{((AU_{n \times k}) \bigodot (AU_{n \times k}) \mathbf{1}_{k \times 1}}$$
(6.9)

Where C_k is an $n \times 1$ vector containing PCC score of each node. Nodes with highest PCC scores are considered the influential nodes. Similar to spectral clustering, a differential private matrix \hat{A} of social network A is computed using Algorithm 3. The published matrix is then used to compute eigenvectors and obtain the PCC scores \hat{C}_k using Equation 6.9.

We evaluate the utility preservation of the published data by evaluating the accuracy with which influential nodes with high ranks are identified. First, for a given value of k, eigenvectors corresponding to the k largest eigenvalues are computed from the original adjacency matrix and the published data i.e., after applying matrix randomization using Algorithm 3. We then obtain PCC scores of all the nodes in a graph from original and published data (denoted as C_k and \hat{C}_k respectively). The original scores C_k and the published scores \hat{C}_k are then compared in two different ways. For all experiments, we compute PCC scores by varying the number of eigenvectors in the range k = 2, 4, 8, 16.

In the first evaluation, we use Mean Square Error (MSE) to compute the error between score values of C_k and \hat{C}_k . We report $n \times MSE$ in our study in order to alleviate the scaling factor induced by the size of social networks. In the second evaluation, we identify two sets of top t influential nodes based on the PCC scores computed from the original data as well as from the published data. We then evaluate the performance of our algorithm by measuring the percentage of overlapped nodes between these two sets. Table 6.4 gives the values of Mean Square Error between PCC scores obtained from the original and published data. We also compare these results with the LNPP approach. For comparison, we show in Table 6.5 the MSE results for baseline LNPP. It is clear that the proposed algorithm yields significantly more accurate estimation of PCC scores than LNPP.

In the second evaluation, we measure the percentage of nodes correctly identified as the top-t influential nodes. First, we obtain two sets T and \hat{T} that contain the top-t most influential nodes measured by the PCC scores given by C_k and \hat{C}_k . Then the percentage of

# of Eigenvectors	2	4	8	16
Facebook	$2.6e^{-26}$	$2.9e^{-4}$	0.021	0.013
Live Journal	$4.0e^{-4}$	0.006	0.034	0.719
Pokec	$3.0e^{-4}$	0.005	0.009	0.019

Table 6.4 $n \times MSE$ using the proposed approach

# of Eigenvectors	2	4	8	16
Facebook	1.83	1.83	1.67	1.64
Live Journal	1.96	1.96	1.88	1.92
Pokec	1.79	1.63	1.62	1.55

Table 6.5 $n \times MSE$ using baseline LNPP

nodes common to both T and \hat{T} is computed. We consider top 10, 100, 1000 and 10000 ranked nodes. Figure 6.8 shows the percentage of nodes correctly identified as the top-t influential nodes for the three datasets. We can see that for all cases, the proposed algorithm is able to recover at least 80% of the most influential nodes. In contrast, LNPP fails to preserve the most influential nodes as the percentage of nodes correctly identified as the top-t influential nodes is less than 1% for all cases.

6.5 Conclusions

In this chapter, we propose a random matrix approach to OSN graph publishing, which achieves space efficiency by reducing the dimensions of adjacency matrices and achieves differential privacy by adding a small amount of noise. The key advantage of our work over prior art is that our proposed random matrix approach achieves both space efficiency and utility preservation, whereas prior work either loses space efficiency (due to outputting large $n \times n$ matrices) or utility preservation (due to the addition of a large amount of noise). We formally prove that our scheme achieves differential privacy and that the amount of added noise is small. Furthermore, we validated our random matrix approach on two different applications: node clustering, and node ranking, which require the spectral information of a graph. Our experimental results show that even for high values of noise variance $\sigma = 1$, our scheme preserves the utility of the dataset.

8 Conclusion

In this thesis, I presented network systems that solve network management needs of three types of operational networks namely; cellular networks, content delivery networks, and online social networks. For cellular networks, I presented an approach to modeling of endto-end network performance. I used statistical, data mining and machine learning methods to develop models for detection and localization of network performance degradation issues. For content delivery networks (CDNs), I presented approaches to actively collect client perceived network performance and use these measurements along with server side logs for modeling and analysis for quantification and optimization of CDN performance. The solutions that were developed have been tested using real data which is collected at operational networks. For online social networks, my research is on developing privacy preserving approaches for graph data publishing.

The vision of this thesis can be extended to many other similar research directions. Network management involves monitoring other important set of metrics that measure end-user QoE, that is measured in terms of qualitative metrics such as user engagement and abandonment rate, and quantitative metrics which are specific to an application e.g., page load time for web browsing and buffering ratio for video streaming. The modeling and analysis framework adopted in this thesis can be leveraged to develop network management systems that automate the process of debugging, quantifying, and optimizing QoE for operational networks.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Alexa Top 500 Global Sites. http://www.alexa.com/topsites.
- [2] Being good stewards of the Internet, https://www.verizondigitalmedia.com/blog/2015/02/being-good-stewards-of-the-internet/.
- [3] Facts & Figures Akamai. http://www.akamai.com/html/about/facts_figures. html.
- [4] Facts & Figures Akamai Technologies, Inc. http://www.akamai.com/html/about/ facts_figures.html.
- [5] Four Ways to Avoid Cascading Failures on a CDN, https://www.verizondigitalmedia.com/blog/2017/06/four-ways-to-avoid-cascadingfailures-on-a-cdn/.
- [6] Google Global Cache (GGC). https://peering.google.com/about/ggc.html.
- [7] Mark the spot. http://itunes.apple.com/us/app/at-t-mark-thespot/id338307313?mt=8.
- [8] Netflix Open Connect Content Delivery for ISPs. https://openconnect.netflix. com.
- [9] PeeringDB. http://www.peeringdb.com.
- [10] Root wireless inc. http://www.rootwireless.com.
- [11] Stanford large network dataset collection. http://snap.stanford.edu/data/.
- [12] White paper: Cisco visual networking index forecast and methodology 2015-2020. Technical report, Cisco.
- [13] Internet Filter: Internet Pornography Statistics. http://internet-filterreview.toptenreviews.com/internet-pornography-statistics.html, 2006.
- [14] Porn Sites Get More Visitors Each Month Than Netflix, Amazon And Twitter Combined. http://www.huffingtonpost.com/2013/05/03/internet-porn-stats_ n_3187682.html, 2013.
- [15] 3GPP. General packet radio service (gprs); gprs tunnelling protocol (gtp) across the gn and gp interface. TS 29.060, 3rd Generation Partnership Project (3GPP), 2011.
- [16] Gaurav Aggarwal, Elie Bursztein, Collin Jackson, and Dan Boneh. An analysis of private browsing modes in modern browsers. In USENIX Security Symposium, 2010.

- [17] F. Ahmed, J. Erman, Z. Ge, A. X. Liu, J. Wang, and H. Yan. Detecting and localizing end-to-end performance degradation for cellular data services based on tcp loss ratio and round trip time. *IEEE/ACM Transactions on Networking*, 2017.
- [18] F. Ahmed, J. Erman, Z. Ge, A. X. Liu, J. Wang, and H. Yan. Monitoring quality-ofexperience for operational cellular networks using machine-to-machine traffic. In 36th Annual IEEE International Conference on Computer Communications (INFOCOM), 2017.
- [19] F. Ahmed, A. X. Liu, and R. Jin. Social graph publishing with privacy guarantees. In *IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 2016.
- [20] F. Ahmed, M. Z. Shafiq, A. R. Khakpour, and A. X. Liu. Optimizing internet transit routing for content delivery networks. *IEEE/ACM Transactions on Networking*, 2017.
- [21] F. Ahmed, M. Z. Shafiq, and A. X. Liu. The internet is for porn: Measurement and analysis of online adult traffic. In *IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 2016.
- [22] F. Ahmed, Z. Shafiq, A. Khakpour, and A. X. Liu. Optimizing internet transit routing for content delivery networks. In *IEEE 24th International Conference on Network Protocols (ICNP)*, 2016.
- [23] Faraz Ahmed, Jeffrey Erman, Zihui Ge, Alex X. Liu, Jia Wang, and He Yan. Detecting and localizing end-to-end performance degradation for cellular data services. In ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS), 2015.
- [24] Faraz Ahmed, Jeffrey Erman, Zihui Ge, Alex X. Liu, Jia Wang, and He Yan. Detecting and localizing end-to-end performance degradation for cellular data services. In 35th Annual IEEE International Conference on Computer Communications (INFOCOM), 2016.
- [25] Aditya Akella, Bruce Maggs, Srinivasan Seshan, Anees Shaikh, and Ramesh Sitaraman. A measurement-based analysis of multihoming. In Proceedings of SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pages 353–364, 2003.
- [26] Isabel Amigo, Pablo Belzarena, and Sandrine Vaton. On the problem of revenue sharing in multi-domain federations. In *Proceedings of IFIP Networking*, pages 252–264, 2012.
- [27] C. Amrutkar, M. Hiltunen, T. Jim, K. Joshi, O. Spatscheck, P. Traynor, and S. Venkataraman. Why is My Smartphone Slow? On the Fly Diagnosis of Underperformance

on the Mobile Internet. In 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pages 1–8, 2013.

- [28] Xueli An and Gerald Kunzmann. Understanding mobile Internet usage behavior. In *IFIP Networking*, 2014.
- [29] M. Austin, J. Fix, S. Meredith, S. Puthenpura, and G. Meempat. Location estimation of a mobile device in a umts network, 2012. US Patent App. 12/870,254.
- [30] AvenueQ. The Internet Is for Porn. https://www.youtube.com/watch?v= LTJvdGcb7Fs.
- [31] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In Proceedings of 16th International Conference of on World Wide Web, pages 181–190. ACM, 2007.
- [32] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A Maltz, and Ming Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. 37(4):13–24, 2007.
- [33] Dziugas Baltrunas, Ahmed Elmokashfi, and Amund Kvalbein. Measuring the reliability of mobile broadband networks. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 45–58. ACM, 2014.
- [34] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnsonlindenstrauss transform itself preserves differential privacy. In Proceedings of 53rd IEEE Annual Symposium on Foundations of Computer Science, pages 410–419. IEEE, 2012.
- [35] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In Proceedings of twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 128–138. ACM, 2005.
- [36] Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120, 1972.
- [37] I. Castro and S. Gorinsky. T4P: Hybrid interconnection for cost reduction. In IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pages 178–183, 2012.
- [38] Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of 18th International Conference of on World Wide Web*, pages 721–730. ACM, 2009.

- [39] Nikolaos Chatzis, Georgios Smaragdakis, Anja Feldmann, and Walter Willinger. There is more to IXPs than meets the eye. ACM SIGCOMM Computer Communication Review, 43(5):19–28, 2013.
- [40] Kamalika Chaudhuri, Anand Sarwate, and Kaushik Sinha. Near-optimal differentially private principal components. In Advances in Neural Information Processing Systems 25, pages 998–1006, 2012.
- [41] M. Chen, E. Kiciman, E. Fratkin, A. Fox, and E. Brewer. Pinpoint: Problem determination in large, dynamic Internet services. In *Proceedings of International Conference* of on Dependable Systems and Networks, pages 595–604, 2002.
- [42] Yu-Chung Cheng, John Bellardo, Péter Benkö, Alex C Snoeren, Geoffrey M Voelker, and Stefan Savage. Jigsaw: solving the puzzle of enterprise 802.11 analysis, volume 36. 2006.
- [43] Y. Crama and P.L. Hammer. Boolean Models and Methods in Mathematics, Computer Science, and Engineering. 2010.
- [44] Amogh Dhamdhere and Constantinos Dovrolis. ISP and egress path selection for multihomed networks. In *Proceedings of IEEE INFOCOM*, pages 1–12, 2006.
- [45] Bowei Du and Michael Demmerand Eric Brewer. Analysis of WWW Traffic in Cambodia and Ghana. In WWW, 2006.
- [46] William Dumouchel and Fanny O'Brien. Integrating a robust option into a multiple regression computing environment. In Proceedings of 21st Symposium on the Interface of Computer Science and Statistics, page 41, 1991.
- [47] Cynthia Dwork. Differential privacy: A survey of results. In Theory and Applications of Models of Computation, pages 1–19. Springer, 2008.
- [48] Jeffrey Erman, Alexandre Gerber, K.K. Ramakrishnan, Subhabrata Sen, and Oliver Spatscheck. Over The Top Video: The Gorilla in Cellular Networks. In ACM Internet Measurement Conference (IMC), 2011.
- [49] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. RFC 2616: Hypertext Transfer Protocol – HTTP/1.1. Technical report, Network Working Group, Internet Engineering Task Force, 1999.
- [50] Alessandro Finamore, Marco Mellia, Maurizio M Munafò, Ruben Torres, and Sanjay G Rao. Youtube everywhere: Impact of device and infrastructure synergies on user experience. In *Proceedings of ACM IMC*, pages 345–360, 2011.

- [51] Benjamin Frank, Ingmar Poese, Yin Lin, Georgios Smaragdakis, Anja Feldmann, Bruce Maggs, Jannis Rake, Steve Uhlig, and Rick Weber. Pushing cdn-isp collaboration to the limit. ACM SIGCOMM Computer Communication Review, 43(3):34–44, 2013.
- [52] Benjamin Frank, Ingmar Poese, Georgios Smaragdakis, Anja Feldmann, Bruce M. Maggs, Steve Uhlig, Vinay Aggarwal, and Fabian Schneider. *Recent Advances in Networking*, chapter Collaboration Opportunities for Content Delivery and Network Infrastructures. ACM SIGCOMM, 2013.
- [53] Eugene C. Freuder and Mark Wallace. Constraint Programming, pages 369–401. Springer US, 2014.
- [54] Ruomei Gao, Constantinos Dovrolis, and Ellen W Zegura. Avoiding oscillations due to intelligent route control systems. In *Proceedings of IEEE INFOCOM*, pages 1–12, 2006.
- [55] Aaron Gember, Aditya Akella, Jeffrey Pang, Alexander Varshavsky, and Ramon Caceres. Obtaining in-context measurements of cellular network performance. In Proceedings of ACM conference on Internet measurement conference, pages 287–300. ACM, 2012.
- [56] David K Goldenberg, Lili Qiuy, Haiyong Xie, Yang Richard Yang, and Yin Zhang. Optimizing cost and performance for multihoming. ACM SIGCOMM Computer Communication Review, 34(4):79–92, 2004.
- [57] Ilya Grigorik. Breaking the 1000 ms Time to Glass Mobile Barrier. In SF HTML5 meetup, 2013.
- [58] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *Proceedings of 13th International Conference* of on World Wide Web, pages 491–501, 2004.
- [59] Silviu Guiaşu. Information theory with applications. McGraw-Hill New York, 1977.
- [60] Laszlo Gyarmati, Rade Stanojevic, Michael Sirivianos, and Nikolaos Laoutaris. Sharing the cost of backbone networks: cui bono? In *Proceedings of ACM IMC*, pages 509–522, 2012.
- [61] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.
- [62] Prashanth Hande, Mung Chiang, Robert Calderbank, and Junshan Zhang. Pricing under constraints in access networks: Revenue maximization and congestion management. In *Proceedings of IEEE INFOCOM*, pages 1–9, 2010.

- [63] Saul Hansell. Aol removes search data on vast group of web users. New York Times, 8, 2006.
- [64] Moritz Hardt and Aaron Roth. Beyond worst-case analysis in private singular vector computation. arXiv preprint arXiv:1211.0975, 2012.
- [65] S. Hasan, S. Gorinsky, C. Dovrolis, and R. K. Sitaraman. Trade-offs in optimizing the cache deployments of CDNs. In *Proceedings of IEEE INFOCOM*, pages 460–468, 2014.
- [66] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *Proceedings of 9th IEEE International Conference of on Data Mining*, pages 169–178. IEEE, 2009.
- [67] Matt Osinski Jennifer Yates He Yan, Zihui Ge. When cell towers fail: Quantifying the customer impact. 2013.
- [68] Cornelis Hoede. A new status score for actors in a social network. Twente University Department of Applied Mathematics (Memorandum no. 243), 1978.
- [69] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. Communications in Statistics-Theory and Methods, 6(9):813–827, 1977.
- [70] Cheng Huang, Angela Wang, Jin Li, and Keith W. Ross. Measuring and evaluating large-scale CDNs. In *Proceedings of ACM IMC*, pages 15–29, 2008.
- [71] Junxian Huang, Feng Qian, Alexandre Gerber, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of 10th International Conference of on MobiSys*, pages 225–238, 2012.
- [72] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. An in-depth study of lte: Effect of network protocol and application behavior on performance. In *Proceedings of ACM SIGCOMM*, pages 363–374, 2013.
- [73] Junxian Huang, Qiang Xu, Birjodh Tiwana, Z Morley Mao, Ming Zhang, and Paramvir Bahl. Anatomizing application performance differences on smartphones. In *Proceedings* of of the 8th International Conference of on Mobile systems, applications, and services, pages 165–178, 2010.
- [74] Qi Huang, Ken Birman, Robbert van Renesse, Wyatt Lloyd, Sanjeev Kumar, and Harry C. Li. An Analysis of Facebook Photo Caching. In SOSP, 2013.
- [75] Muhammad U Ilyas and Hayder Radha. Identifying influential nodes in online social networks using principal component centrality. In *Proceedings of IEEE International Conference of on Communications*, pages 1–5. IEEE, 2011.

- [76] Muhammad U Ilyas, M Zubair Shafiq, Alex X Liu, and Hayder Radha. A distributed and privacy preserving algorithm for identifying information hubs in social networks. In *INFOCOM*, 2011 Proceedings IEEE, pages 561–565. IEEE, 2011.
- [77] Libin Jiang, Shyam Parekh, and Jean Walrand. Time-dependent network pricing and bandwidth trading. In Proceedings of IEEE Network Operations and Management Symposium Workshops, NOMS, pages 193–200, 2008.
- [78] Yu Jin, Nick Duffield, Alexandre Gerber, Patrick Haffner, Wen-Ling Hsu, Guy Jacobson, Subhabrata Sen, Shobha Venkataraman, and Zhi-Li Zhang. Large-scale app-based reporting of customer problems in cellular networks: Potential and limitations. In Proceedings of ACM SIGCOMM workshop on Measurements up the stack, pages 13–18. ACM, 2011.
- [79] Diana Joumblatt, Jaideep Chandrashekar, Branislav Kveton, Nina Taft, and Renata Teixeira. Predicting user dissatisfaction with internet application performance at endhosts. In *Proceedings of INFOCOM*, pages 235–239. IEEE, 2013.
- [80] S. Kandula, D. Katabi, and J.P. Vasseur. Shrink: A tool for failure diagnosis in IP networks. In *Proceedings of ACM SIGCOMM workshop on Mining network data*, pages 173–178, 2005.
- [81] Srikanth Kandula, Ratul Mahajan, Patrick Verkaik, Sharad Agarwal, Jitendra Padhye, and Paramvir Bahl. Detailed diagnosis in enterprise networks. *Proceedings of ACM SIGCOMM*, pages 243–254, 2009.
- [82] Michael Kapralov, Kunal Talwar, Michael Kapralov, and Kunal Talwar. On differentially private low rank approximation. In Proc. 24rd Symposium on Discrete Algorithms (SODA), 2012.
- [83] Richard M. Karp. Reducibility among Combinatorial Problems. Springer US, Boston, MA, 1972.
- [84] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
- [85] Leonard Kaufman and Peter J Rousseeuw. Finding groups in data: an introduction to cluster analysis, volume 344. John Wiley & Sons, 2009.
- [86] Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the johnson-lindenstrauss transform. arXiv preprint arXiv:1204.2606, 2012.
- [87] George Kesidis, Apurba Das, and Gustavo de Veciana. On flat-rate and usage-based pricing for tiered commodity internet services. In *Proceedings of IEEE CISS*, pages 304–308, 2008.

- [88] I.Y. Kim and O.L. de Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 29(2):149–158, 2005.
- [89] Ramana Rao Kompella, Jennifer Yates, Albert Greenberg, and Alex C. Snoeren. Ip fault localization via risk modeling. In *Proceedings of NSDI*, pages 57–70, 2005.
- [90] RR Kompella, J. Yates, A. Greenberg, and AC Snoeren. Detection and localization of network black holes. In *Proceedings of IEEE INFOCOM*, pages 2180–2188, 2007.
- [91] Rupa Krishnan, Harsha V. Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, and Jie Gao. Moving beyond end-to-end path information to optimize CDN performance. In *Proceedings of ACM IMC*, pages 190– 201, 2009.
- [92] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of 19th International Conference of* on World wide web, pages 591–600. ACM, 2010.
- [93] Craig Labovitz. First Data on Changing Netflix and CDN Market Share, June 2012. http://www.deepfield.net/2012/06/first-data-on-changing-netflix-and-cdnmarket-share/.
- [94] Markus Laner, Philipp Svoboda, Stefan Schwarz, and Markus Rupp. Users in cells: a data traffic analysis. In Wireless Communications and Networking Conference (WCNC), pages 3063–3068. IEEE, 2012.
- [95] Andrew Lipsman. Major Mobile Milestones in May: Apps Now Drive Half of All Time Spent on Digital. http://www.comscore.com/Insights/Blog/ Major-Mobile-Milestones-in-May-Apps-Now-Drive-Half-of-All-Time-Spent-on-Digital, June 2014.
- [96] A. Lodhi, N. Larson, A. Dhamdhere, C. Dovrolis, and kc claffy. Using PeeringDB to understand the peering ecosystem. ACM SIGCOMM Computer Communication Review, 44(2):20-27, 2014.
- [97] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Mining social networks using heat diffusion processes for marketing candidates selection. In Proceedings of 17th ACM Conference of on Information and knowledge management, pages 233–242. ACM, 2008.
- [98] Ajay Anil Mahimkar, Zihui Ge, Aman Shaikh, Jia Wang, Jennifer Yates, Yin Zhang, and Qi Zhao. Towards automated performance diagnosis in a large iptv network. In *Proceedings of ACM SIGCOMM*, pages 231–242, 2009.

- [99] Ajay Anil Mahimkar, Han Hee Song, Zihui Ge, Aman Shaikh, Jia Wang, Jennifer Yates, Yin Zhang, and Joanne Emmons. Detecting the performance impact of upgrades in large operational networks. ACM SIGCOMM Computer Communication Review, pages 303–314, 2011.
- [100] Mahesh K Marina, Valentin Radu, and Konstantinos Balampekos. Impact of indooroutdoor context on crowdsourcing based mobile coverage analysis. In Proceedings of Workshop on All Things Cellular: Operations, Applications and Challenges, pages 45–50. ACM, 2015.
- [101] Darakhshan Mir and Rebecca N Wright. A differentially private estimator for the stochastic kronecker graph model. In *Proceedings of Joint EDBT/ICDT Workshops*, pages 167–176. ACM, 2012.
- [102] Meinard Müller. Dynamic Time Warping. Information retrieval for music and motion, pages 69–84, 2007.
- [103] Diala Naboulsi, Marco Fiore, Stephane Ribot, and Razvan Stanica. Large-scale Mobile Traffic Analysis: a Survey. *IEEE Communications Surveys & Tutorials*, 2015.
- [104] Srinivas Narayana, Wenjie Jiang, Jennifer Rexford, and Mung Chiang. Joint server selection and routing for geo-replicated services. In *Proceedings of IEEE/ACM Conference of on Utility and Cloud Computing*, pages 423–428, 2013.
- [105] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In Proceedings of IEEE Symposium on Security and Privacy, pages 111–125. IEEE, 2008.
- [106] Ashkan Nikravesh, David R Choffnes, Ethan Katz-Bassett, Z Morley Mao, and Matt Welsh. Mobile network performance from user devices: A longitudinal, multidimensional analysis. In *Passive and Active Measurement*, pages 12–22. Springer, 2014.
- [107] W. B. Norton. The 2014 Internet Peering Playbook: Connecting to the Core of the Internet. DrPeering Press, 2014.
- [108] William B Norton. The Internet Peering Playbook: Connecting to the Core of the Internet. DrPeering Press, 2011.
- [109] Ogi Ogas and Sai Gaddam. A billion wicked thoughts: What the world's largest experiment reveals about human desire. Dutton New York, NY, 2011.
- [110] OpenSignal. http://opensignal.com/reports/fragmentation-2013/.
- [111] Mukaddim Pathan and Rajkumar Buyya. Content Delivery Networks, Chapter 2: A Taxonomy of CDNs. Springer, 2008.

- [112] Utpal Paul, Anand Prabhu Subramanian, Milind Madhav Buddhikot, and Samir R. Das. Understanding Traffic Dynamics in Cellular Data Networks. In *IEEE Infocom*, 2011.
- [113] Davide Proserpio, Sharon Goldberg, and Frank McSherry. A workflow for differentiallyprivate graph synthesis. In *Proceedings of ACM Workshop on online social networks*, pages 13–18. ACM, 2012.
- [114] Feng Qian, Zhaoguang Wang, Alexandre Gerber, Zhuoqing Mao, Subhabrata Sen, and Oliver Spatscheck. Profiling resource usage for mobile applications: a cross-layer approach. In *Proceedings of 9th MobiSys*, pages 321–334, 2011.
- [115] Lili Qiu, V. N. Padmanabhan, and G. M. Voelker. On the placement of Web server replicas. In *Proceedings of IEEE INFOCOM*, pages 1587–1596, 2001.
- [116] Prabhakar Raghavan and Clark D Tompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.
- [117] Fabio Ricciato. Traffic monitoring and analysis for the optimization of a 3g network. Wireless Communications, IEEE, pages 42–49, 2006.
- [118] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In Proceedings of 8th ACM SIGKDD International Conference of on Knowledge Discovery and Data Mining, pages 61–70. ACM, 2002.
- [119] Tiago Rodrigues, Fabrcio Benevenuto, Meeyoung Cha, Krishna P. Gummadi, and Virgilio Almeida. On Word-of-Mouth Based Discovery of the Web. In ACM Internet Measurement Conference (IMC), 2011.
- [120] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y Zhao. Sharing graphs using differentially private graph models. In *Proceedings of ACM SIG-COMM Internet Measurement Conference of*, pages 81–98. ACM, 2011.
- [121] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS '06, pages 143–152, 2006.
- [122] Alexander Schrijver. Theory of Linear and Integer Programming. John Wiley & Sons, 1998.
- [123] Sayandeep Sen, Jongwon Yoon, Joshua Hare, Justin Ormont, and Suman Banerjee. Can they hear me now?: a case for a client-assisted approach to monitoring widearea wireless networks. In *Proceedings of ACM SIGCOMM conference on Internet measurement conference*, pages 99–116. ACM, 2011.

- [124] M. Zubair Shafiq, Jeffrey Erman, Lusheng Ji, Alex X. Liu, Jeffrey Pang, and Jia Wang. Understanding the Impact of Network Dynamics on Mobile Video User Engagement. In ACM SIGMETRICS, 2014.
- [125] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, and Jia Wang. Characterizing and modeling internet traffic dynamics of cellular devices. In *Proceedings of ACM SIGMETRICS*, pages 305–316, San Jose, California, June 2011.
- [126] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, and Jia Wang. Characterizing and Modeling Internet Traffic Dynamics of Cellular Devices. In ACM SIGMETRICS, 2011.
- [127] Joel Sommers and Paul Barford. Cell vs. wifi: on the performance of metro area mobile connections. In *Proceedings of ACM conference on Internet measurement conference*, pages 301–314. ACM, 2012.
- [128] Han Hee Song, Zihui Ge, Ajay Mahimkar, Jia Wang, Jennifer Yates, Yin Zhang, Andrea Basso, and Min Chen. Q-score: proactive service quality assessment in a large iptv system. In *Proceedings of ACM SIGCOMM conference on Internet measurement* conference, pages 195–208. ACM, 2011.
- [129] Sebastian Sonntag, Lennart Schulte, and Jukka Manner. Mobile network measurements-it's not all about signal strength. In Proceedings of Wireless Communications and Networking Conference (WCNC), pages 4624–4629. IEEE, 2013.
- [130] Rade Stanojevic, Ignacio Castro, and Sergey Gorinsky. CIPT: Using tuangou to reduce IP transit costs. In *Proceedings of ACM CoNEXT*, page 17, 2011.
- [131] Rade Stanojevic, Nikolaos Laoutaris, and Pablo Rodriguez. On economic heavy hitters: Shapley value analysis of 95th-percentile pricing. In *Proceedings of ACM IMC*, pages 75–80, 2010.
- [132] Ao-Jan Su, David R Choffnes, Aleksandar Kuzmanovic, and Fabián E Bustamante. Drafting behind Akamai: Inferring network conditions based on CDN redirections. *IEEE/ACM Transactions on Networking (TON)*, 17(6):1752–1756, 2009.
- [133] Peng Sun, Minlan Yu, Michael J Freedman, and Jennifer Rexford. Identifying performance bottlenecks in CDNs through TCP-level monitoring. In Proceedings of ACM SIGCOMM workshop on Measurements up the stack, pages 49–54, 2011.
- [134] Shu Tao, Kuai Xu, Ying Xu, Teng Fei, Lixin Gao, Roch Guerin, Jim Kurose, Don Towsley, and Zhi Li Zhang. Exploring the performance benefits of end-to-end path switching. In *Proceedings of IEEE ICNP*, pages 304–315, 2004.
- [135] Gareth Tyson, Yehia Elkhatib, Nishanth Sastry, and Steve Uhlig. Demystifying Porn 2.0: A look into a major adult video streaming website. In ACM Internet Measurement Conference (IMC), pages 417–426. ACM, 2013.

- [136] Gareth Tyson, Yehia Elkhatib, Nishanth Sastry, and Steve Uhlig. Are People Really Social on Porn 2.0? In AAAI Conference on Web and Social Media (ICWSM), 2015.
- [137] Vytautas Valancius, Cristian Lumezanu, Nick Feamster, Ramesh Johari, and Vijay V Vazirani. How many tiers?: Pricing in the Internet transit market. In *Proceedings of* ACM SIGCOMM, pages 194–205, 2011.
- [138] R.J. Vanderbei. Linear Programming: Foundations and Extensions. International Series in Operations Research & Management Science. 2007.
- [139] Deepak Verma and Marina Meila. A comparison of spectral clustering algorithms. Technical report, 2003.
- [140] Hao Wang, Haiyong Xie, Lei Qiu, Avi Silberschatz, and Yang Richard Yang. Optimal ISP subscription for Internet multihoming: Algorithm design and implication analysis. In *Proceedings of IEEE INFOCOM*, pages 2360–2371, 2005.
- [141] Yue Wang, Xintao Wu, and Leting Wu. Differential privacy preserving spectral graph analysis. In Advances in Knowledge Discovery and Data Mining, volume 7819 of Lecture Notes in Computer Science, pages 329–340. Springer Berlin Heidelberg, 2013.
- [142] Mark Ward. Web porn: Just how much is there? http://www.bbc.com/news/ technology-23030090, July 2013.
- [143] C. Wilson, B. Boe, A. Sala, K.P.N. Puttaswamy, and B.Y. Zhao. User interactions in social networks and their implications. In *Proceedings of ACM European Conference* on Computer Systems, 2009.
- [144] Gilbert Wondracek, Thorsten Holz, Christian Platzer, Engin Kirda, and Christopher Kruegel. Is the Internet for Porn? An Insight Into the Online Adult Industry. In Workshop on the Economics of Information Security (WEIS), 2010.
- [145] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. Identifying diverse usage behaviors of smartphone apps. In ACM Internet Measurement Conference (IMC), 2011.
- [146] xvideos.com Site Overview Alexa. http://www.alexa.com/siteinfo/xvideos.com.
- [147] youtube.com Site Overview Alexa. http://www.alexa.com/siteinfo/youtube.com.
- [148] Ying Zhang and Ake Arvidsson. Understanding the characteristics of cellular data traffic. ACM SIGCOMM Computer Communication Review, 42(4):461–466, 2012.
- [149] Zheng Zhang, Ming Zhang, Albert G Greenberg, Y Charlie Hu, Ratul Mahajan, and Blaine Christian. Optimizing cost and performance in online service provider networks. In *Proceedings of USENIX NSDI*, pages 33–48, 2010.